



## Änderungsdetektion digitaler Fernerkundungsdaten mittels objekt-basierter Bildanalyse

Clemens Listner







Forschungszentrum Jülich GmbH  
Institut für Energie- und Klimaforschung  
Nukleare Entsorgung und Reaktorsicherheit (IEK-6)

# Änderungsdetektion digitaler Fernerkundungs- daten mittels objekt-basierter Bildanalyse

Clemens Listner

Schriften des Forschungszentrums Jülich  
Reihe Energie & Umwelt / Energy & Environment

Band / Volume 242

---

ISSN 1866-1793

ISBN 978-3-95806-017-3

Bibliografische Information der Deutschen Nationalbibliothek.  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der  
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten  
sind im Internet über <http://dnb.d-nb.de> abrufbar.

Herausgeber  
und Vertrieb: Forschungszentrum Jülich GmbH  
Zentralbibliothek, Verlag  
52425 Jülich  
Tel.: +49 2461 61-5368  
Fax: +49 2461 61-6103  
E-Mail: [zb-publikation@fz-juelich.de](mailto:zb-publikation@fz-juelich.de)  
[www.fz-juelich.de/zb](http://www.fz-juelich.de/zb)

Umschlaggestaltung: Grafische Medien, Forschungszentrum Jülich GmbH

Druck: Grafische Medien, Forschungszentrum Jülich GmbH

Copyright: Forschungszentrum Jülich 2014

Schriften des Forschungszentrums Jülich  
Reihe Energie & Umwelt / Energy & Environment, Band / Volume 242

D 105 (Diss., TU Bergakademie Freiberg, 2014)

ISSN 1866-1793

ISBN 978-3-95806-017-3

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)  
unter [www.fz-juelich.de/zb/openaccess](http://www.fz-juelich.de/zb/openaccess).

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie oder  
in einem anderen Verfahren) ohne schriftliche Genehmigung des Verlages reproduziert oder  
unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

## Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurde ein Verfahren zur Detektion von Änderungen auf der Erdoberfläche anhand digitaler Satellitendaten entwickelt, implementiert und getestet. Im Unterschied zu bestehenden Verfahren konnte das Paradigma der objekt-basierten Bildanalyse vollständig umgesetzt werden.

Kern der Methode ist ein Algorithmus zur Extraktion von Bildobjekten, der bitemporal stabile Ergebnisse liefert. Darüber hinaus präsentiert die Arbeit sowohl neue Verfahren zur Verknüpfung der Objekte beider Aufnahmen als auch für die Auswahl der zu verwendenden Objekteigenschaften. Numerische Probleme der IR-MAD-Methode bei der objekt-basierten Änderungsdetektion konnten im Rahmen dieser Arbeit ausgeräumt werden. Für die anschließende Änderungsklassifizierung kommen erweiterte Klassifikationsverfahren zum Einsatz.

Alle Teilmethoden wurden innerhalb der Software *eCognition* integriert. Das Verfahren wurde sowohl auf simulierte als auch auf reale Daten angewandt. Die Ergebnisse wurden mit jenen alternativer Bildverarbeitungsansätze verglichen. Teilweise konnten quantitativ messbare Verbesserungen der Güte erreicht werden.



# Abstract

Within the framework of this thesis, a procedure has been developed, implemented and tested which is capable of detecting changes on the Earth's surface using digital satellite data. In contrast to existing methods, this procedure fully implements the paradigm of object-based image analysis (OBIA).

The core of the procedure is an algorithm for the extraction of objects from images, also known as segmentation, which gives bitemporally stable results. Moreover, new methods for linking objects between two segmentations are presented and a heuristic is proposed for object feature selection. Numerical problems in applying the IR-MAD method to object-based change detection have been overcome. Existing classification algorithms for subject change classification were used and extended in this work.

All parts of the object-based change detection procedure were integrated into the image analysis software *eCognition*.

The procedure was applied to simulated as well as to real datasets. The results were compared to those of alternative image processing techniques. In part, the procedure displays measurable improvements in the quality of the results.





## Danksagung

Mein ganz besonderer Dank gilt meiner Doktormutter Frau Dr. Irmgard Niemeyer. Ohne ihr Zutrauen, die fachliche Unterstützung, die Bereitstellung der finanziellen Mittel sowie ihre Geduld würde diese Dissertation heute nicht vorliegen.

Herrn Prof. Helmut Schaeben und Herrn Prof. Oliver Ernst danke ich für Ihre Bereitschaft, die Begutachtung der Dissertation zu übernehmen.

Herrn Prof. Anton Sroka und Herrn Prof. Dirk Bosbach möchte ich danken, dass ich die Dissertation an ihren jeweiligen Instituten anfertigen durfte.

Herrn Dr. Mort Canty gilt mein besonderer Dank für die hilfreichen Diskussionen sowie seine konstruktiven Vorschläge zur Verbesserung des Textes.

Meiner Frau Carolin Listner danke ich besonders für die Motivation und Ihre Geduld während der Promotionszeit.

Bei meinen Eltern möchte ich mich für Ihre moralische und finanzielle Unterstützung während meines gesamten Studiums bedanken.



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>13</b>
<b>Tabellenverzeichnis</b>	<b>17</b>
<b>1. Grundlagen</b>	<b>21</b>
1.1. Einleitung . . . . .	21
1.2. Fernerkundung . . . . .	23
1.3. Aktueller Forschungsstand . . . . .	24
1.3.1. Änderungsdetektion . . . . .	24
1.3.2. Objekt-basierte Bildanalyse . . . . .	26
1.3.3. Objekt-basierte Änderungsdetektion . . . . .	29
1.4. Ziele der Arbeit . . . . .	31
1.5. Struktur der Arbeit . . . . .	32
<b>2. Methoden</b>	<b>35</b>
2.1. Einleitung . . . . .	35
2.2. Vorverarbeitung . . . . .	37
2.3. Segmentierung . . . . .	38
2.3.1. Einleitung . . . . .	38
2.3.2. Verfahren zur Segmentierung von Einzelbildern . . . . .	41
2.3.3. Multiresolution Segmentation . . . . .	51
2.3.4. Bildsegmentierung für die objekt-basierte Änderungsdetektion . . . . .	70
2.3.5. Segmentierung großer Bilder . . . . .	86
2.3.6. Zusammenfassung . . . . .	90
2.4. Merkmalsextraktion . . . . .	90
2.4.1. Einleitung . . . . .	90
2.4.2. Objektentsprechung . . . . .	90
2.4.3. Merkmalsauswahl . . . . .	93
2.4.4. Zusammenfassung . . . . .	96
2.5. Transformation des Merkmalsraumes . . . . .	96
2.5.1. Einleitung . . . . .	96
2.5.2. Multivariate Datenanalyse . . . . .	96
2.5.3. Hauptkomponentenanalyse . . . . .	98
2.5.4. Multivariate Änderungsdetektion . . . . .	103
2.5.5. Zusammenfassung . . . . .	109
2.6. Änderungsklassifizierung . . . . .	110
2.6.1. Einleitung . . . . .	110
2.6.2. Klassifizierung mit Schwellenwerten . . . . .	111
2.6.3. Unüberwachte Klassifizierung mittels automatischem Clustering . . . . .	114
2.6.4. Überwachte Klassifizierung mit neuronalen Netzwerken . . . . .	117
2.6.5. Zusammenfassung . . . . .	118
2.7. Nachverarbeitung . . . . .	119
2.7.1. Einleitung . . . . .	119
2.7.2. Datenreduktion durch Kurvenglättung der Änderungsobjekte . . . . .	119

2.7.3. Zusammenfassung . . . . .	125
2.8. Genauigkeitsuntersuchung . . . . .	125
2.8.1. Einleitung . . . . .	125
2.8.2. Güte der Segmentierung . . . . .	125
2.8.3. Güte der Änderungsklassifikation . . . . .	129
2.8.4. Zusammenfassung . . . . .	132
2.9. Zusammenfassung . . . . .	132
<b>3. Anwendung</b>	<b>133</b>
3.1. Einleitung . . . . .	133
3.2. Fallstudie „Autobahn“ . . . . .	133
3.2.1. Einleitung . . . . .	133
3.2.2. Vorverarbeitung . . . . .	134
3.2.3. Segmentierung . . . . .	134
3.2.4. Merkmalsextraktion . . . . .	135
3.2.5. Transformation des Merkmalsraumes . . . . .	136
3.2.6. Änderungsklassifizierung . . . . .	138
3.2.7. Genauigkeitsuntersuchung . . . . .	138
3.2.8. Diskussion . . . . .	140
3.3. Fallstudie „Olkiluoto“ . . . . .	141
3.3.1. Einleitung . . . . .	141
3.3.2. Vorverarbeitung . . . . .	142
3.3.3. Segmentierung . . . . .	142
3.3.4. Merkmalsextraktion . . . . .	145
3.3.5. Transformation des Merkmalsraumes . . . . .	145
3.3.6. Änderungsklassifizierung . . . . .	149
3.3.7. Genauigkeitsuntersuchung . . . . .	155
3.3.8. Diskussion . . . . .	156
3.4. Zusammenfassung der Ergebnisse . . . . .	157
<b>4. Zusammenfassung und Ausblick</b>	<b>159</b>
4.1. Zusammenfassung . . . . .	159
4.2. Ausblick . . . . .	161
<b>A. Simulation von Bildpaaren</b>	<b>163</b>
<b>Literatur</b>	<b>165</b>

# Abbildungsverzeichnis

1.1. Strahlungsmodell der Fernerkundung aus Albers (2009).	23
2.1. Ablauf der objekt-basierten Änderungsdetektion.	35
2.2. Ausgabe der Anwendung des <i>Multiresolution-Segmentation</i> -Algorithmus auf zwei Bilder, die sich nur durch additives Gaußsches Rauschen ( $\sigma = 0.5$ ) unterscheiden.	41
2.3. Strategie des <i>Local Mutual Best Fitting</i> .	51
2.4. Klassendiagramm zur Implementierung des <i>Multiresolution-Segmentation</i> -Algorithmus.	55
2.5. Vergleich der Ergebnisse des <i>Multiresolution-Segmentation</i> -Algorithmus der Originalimplementierung mit der reimplementierten Version.	60
2.6. Abhängigkeit der resultierenden Objektanzahl $m$ vom Heterogenitätsschwellenwert $T$ .	64
2.7. Abhängigkeit der Laufzeit von der Eingabelänge $n$ .	65
2.8. Abhängigkeit der Laufzeit von der Kanalanzahl $k$ .	66
2.9. Abhängigkeit der Laufzeit von der Objektanzahl $m$ .	66
2.10. Abhängigkeit des Speicherbedarfs von der Eingabelänge $n$ .	71
2.11. Originalvarianten der Segmentierung für die objekt-basierte Änderungsdetektion mit Parametern $T = 25$ , $w_s = 0.1$ und $w_{comp} = 0.5$ .	72
2.12. Segmentbereinigung mittels <i>Universal Removal Strategy</i> .	75
2.13. Segmentbereinigung mittels <i>Global Removal Strategy</i> .	76
2.14. Segmentbereinigung mittels <i>Local Removal Strategy</i> .	76
2.15. Segmentbereinigung mittels angepasster <i>Local Removal Strategy</i> .	77
2.16. Anwendungsbeispiele der Verfahren zur Bereinigung inkonsistenter Segmente.	78
2.17. Wahl des Parameters $T_l$ anhand der Bilddaten.	79
2.18. Ausschnitt der Eingabemaske des MRS4CD Plugins für <i>eCognition</i> .	79
2.19. Beispielergebnis des MRS4CD Plugins für <i>eCognition</i> .	81
2.20. Konzept zur Speicherung der Nachbarschaftsbeziehungen transienter Segmente.	83
2.21. Problem der Kachelgrenzen bei der direkten Anwendung der <i>Tiling-Stitching</i> -Methode.	86
2.22. Fünf Hierarchieebenen einer Segmentation zweier Kacheln.	88
2.23. Iterative Aufteilung der Grenzsegmente einer Segmentation auf die darunterliegenden Ebenen.	88
2.24. Überblick des TS-Verfahrens und Vergleich mit den Ergebnissen der sequenziellen Methode.	89
2.25. Zwei Methoden zur Lösung des Problems der Objektentsprechung.	91
2.26. Beispiel für die Methode DCD.	92
2.27. Beispiel für die Methode CDIO.	93
2.28. Eingabemaske des Algorithmus zur Entropiebestimmung eines Objektmerkmals.	95
2.29. Entropie einiger Objektmerkmale in Abhängigkeit vom Parameter $T$ des MRS-Algorithmus.	95
2.30. Eingabemaske des Plugins zur Hauptkomponentenanalyse für <i>eCognition</i> .	103



2.31. Beispielergebnis des <i>eCognition</i> -Plugins zur Hauptkomponententransformation unter Verwendung der Objekt-Eigenschaften Mittelwert, Standardabweichung und Schiefe aller drei Farbkanäle des Bildes. . . . .	104
2.32. Eingabemaske des IR-MAD-Plugins für <i>eCognition</i> . . . . .	108
2.33. Beispielergebnis des <i>eCognition</i> -Plugins zur Multivariaten Änderungsdetektion unter Verwendung der Objekt-Eigenschaften Mittelwert und Standardabweichung aller drei Farbkanäle. . . . .	109
2.34. Gemeinsames Histogramm. . . . .	112
2.35. Schwellenwertbestimmung am Schnittpunkt der gewichteten Dichtefunktionen. . . . .	113
2.36. Beispielergebnis der interaktiven Schwellenwertklassifizierung (rechts) anhand des $Z$ -Wertes (links). . . . .	113
2.37. Ausschnitt der Eingabemaske des FMLE-Plugins für <i>eCognition</i> . . . . .	116
2.38. Eingabemaske des <i>eCognition</i> -Plugins zur Klassifizierung mit neuronalen Netzwerken. . . . .	118
2.39. Beispiel einer Linie, deren Digitalisierung zu viele Punkte aufweist. . . . .	120
2.40. Ermittlung des Lotfußpunktes $A_g$ beim Fällen des Lotes von Punkt $A$ auf die Gerade $g$ . . . . .	121
2.41. Beispielausgabe des Douglas-Peucker-Algorithmus für verschiedene Werte des Parameters $\varepsilon$ . . . . .	123
2.42. Beispielanwendung des Douglas-Peucker-Algorithmus mit Zwischenergebnissen. . . . .	124
2.43. Segmentierungsgüte $E$ in Abhängigkeit vom Parameter $T$ . . . . .	130
2.44. Vier Segmentationen mit lokalen Minima des Güteindicators $E$ . . . . .	130
2.45. Beispiel für eine Konfusionsmatrix zur Evaluierung der Klassifizierungsgenauigkeit. . . . .	132
3.1. Für die Änderungsanalyse in Abschnitt 3.2 verwendete Bilddaten. . . . .	134
3.2. Segmentierung mithilfe des MRS4CD-Algorithmus für die Verarbeitungsvarianten A und B. . . . .	135
3.3. Eingabedaten und Ergebnis des MRS-Algorithmus auf Grundlage der pixelbasierten MAD-Variate für die Verarbeitungsvariante C. . . . .	136
3.4. $Z$ -Werte der IR-MAD-Transformation für die einzelnen Varianten. . . . .	137
3.5. Klassifikationsergebnisse für die einzelnen Varianten. . . . .	139
3.6. Manuell erstellte Referenzklassifikation. . . . .	140
3.7. Die beiden zur Analyse verwendeten Satellitenaufnahmen. . . . .	142
3.8. Ausschnitt der Segmentation zum Aufnahmezeitpunkt 16. Juni 2005 mit $T = 39$ , $w_s = 0.2$ und $w_{comp} = 0.8$ . . . . .	143
3.9. Segmentierung mithilfe des MRS4CD-Algorithmus. . . . .	144
3.10. Vier exemplarische Ausschnitte des verwendeten Bilddatensatzes. . . . .	146
3.11. Ausschnitte der Änderungsintensitätsbilder bei Verwendung diverser Segmentierungsvarianten. . . . .	147
3.12. Indikatorbilder für Änderungen zwischen den beiden Aufnahmezeitpunkten. . . . .	148
3.13. $Z$ -Werte des IR-MAD-Algorithmus mit und ohne vorheriger Datenreduktion durch die Hauptkomponentenanalyse. . . . .	149
3.14. $Z$ -Werte des IR-MAD-Algorithmus unter Verwendung verschiedener Objektmerkmale. . . . .	150
3.15. Ergebnis der iterativen Anwendung des FMLE-Algorithmus bei Entfernung des Clusters mit den durchschnittlich geringsten $Z$ -Werten. . . . .	152
3.16. Ergebnisse der Änderungsklassifikation mit Schwellenwertverfahren. . . . .	153
3.17. Verlauf der Gütemaße in Abhängigkeit von der Clusteranzahl $c$ . . . . .	154
3.18. Ergebnis des FMLE-Algorithmus für verschiedene Werte des Parameters $c$ . . . . .	155

---

A.1. Beispielergebnis für die Bildsimulation im Rahmen der objekt-basierten Änderungsdetektion ( $\sigma = 50$ ). . . . .	164
---	-----



# Tabellenverzeichnis

1.1. Satellitenmissionen mit sehr hochauflösenden optischen Sensoren, deren Daten kommerziell erhältlich sind, sortiert nach Startdatum. . . . .	24
2.1. Verfahrensschritte des Segmentierungsalgorithmus <i>Multiresolution Segmentation for Change Detection</i> . . . . .	73
2.2. Exemplarische Übersicht der Laufzeit des Algorithmus und dessen Speicherbedarf.	85
2.3. Vergleich der Güte verschiedener Segmentierungsverfahren. . . . .	128
2.4. Vergleich der Güte verschiedener bitemporaler Segmentierungsverfahren. . . .	129
3.1. Zusammenfassung der Verarbeitungsschritte. . . . .	139
3.2. Genauigkeit der Ergebnisse der Änderungsdetektion. . . . .	140
3.3. Ergebnisse der Schwellenwertauswahl für die bitemporale Segmentierung bei verschiedenen Bereinigungsverfahren und Datengrundlagen. . . . .	144
3.4. Objektmerkmale und zugehörige Entropiewerte absteigend sortiert nach dem Durchschnitt der Entropie. . . . .	145
3.5. Genauigkeit der Ergebnisse nach Verarbeitung mit dem iterativen FMLE-Algorithmus. . . . .	155
3.6. Genauigkeit der Ergebnisse nach Verarbeitung mit der Schwellenwertklassifikation. . . . .	155



# Abkürzungsverzeichnis

CDIO	<i>Change Detection Using Intersected Objects</i>
DCD	<i>Directed Change Detection</i>
ECHO	<i>Extraction and Classification of Homogeneous Objects</i>
EM	<i>Expectation Maximization</i>
FLAASH	<i>Fast Line-Of-Sight Atmospheric Analysis of Spectral Hypercubes</i>
FMLE	<i>Fuzzy Maximum Likelihood Estimation</i>
GEOBIA	<i>Geographic Object-based Image Analysis</i>
GIS	Geographisches Informationssystem
GLCM	<i>Gray-Level Co-Occurrence Matrix</i>
GSD	<i>Ground Sampling Distance</i>
HKA	Hauptkomponentenanalyse
IAEO	Internationalen Atomenergie Organisation
IR-MAD	<i>Iteratively Reweighted Multivariate Alteration Detection</i>
K-S-Test	Kolmogorov-Smirnov-Test
KKA	Kanonische Korrelationsanalyse
LLNHR	<i>Low-Level Neighbourhood Relationships</i>
LLS	<i>Low-level Segments</i>
MAD	Multivariate Änderungsdetektion
MAF	<i>Maximum Autocorrelation Factor Analysis</i>
MRS	<i>Multiresolution Segmentation</i>
MRS4CD	<i>Multiresolution Segmentation for Change Detection</i>
NDVI	<i>Normalized Differenced Vegetation Index</i>
NTLS	<i>Non-Top-Level Segments</i>
NVV	Vertrag über die Nichtverbreitung von Kernwaffen
OBIA	Objekt-basierte Bildanalyse
PCA	<i>Principal Component Analysis</i>
SDK	<i>Software Development Kit</i>



SEaTH	<i>SEparability and THresholds</i>
SMAP	<i>Sequential Maximum A Posteriori</i>
SNR	<i>Signal-To-Noise Ratio</i>
TLNHR	<i>Top-Level Neighbourhood Relationships</i>
TLS	<i>Top-Level Segments</i>
TS	<i>Tiling-Stitching</i>

# 1. Grundlagen

## 1.1. Einleitung

Der militärischen Nutzung der Kernenergie in Form nuklearer Sprengsätze steht die zivile Anwendung unter anderem zur Produktion von medizinisch einsetzbaren Radionukliden sowie zur Stromerzeugung gegenüber. Der Vertrag über die Nichtverbreitung von Kernwaffen (NVV) fördert ausdrücklich diese zivile Nutzung (siehe *Treaty on the Non-Proliferation of Nuclear Weapons* 1968, Art. IV). Gleichzeitig wird im Vertrag versucht, die militärische Nutzung zu unterbinden. Dazu unterscheidet der Vertrag zwischen Kernwaffenstaaten und Nichtkernwaffenstaaten mit einer klaren Trennlinie, welche einerseits besagt, dass kein Kernwaffenstaat Technologien oder Wissen zur Produktion von nuklearen Waffen an Nichtkernwaffenstaaten übermitteln darf (Art. I). Andererseits dürfen Nichtkernwaffenstaaten nukleare Waffen weder produzieren noch besitzen (Art. II). Diese Trennlinie des Vertrages zu überwachen, ist Aufgabe der Internationalen Atomenergie Organisation (IAEO), die dazu sogenannte Safeguardsmaßnahmen einsetzt.

Zur praktischen Umsetzung dieser Safeguardsmaßnahmen werden zwischen der IAEO und dem betreffenden Staat Rahmenverträge gemäß des Modellabkommens INFCIRC/153 (siehe Internationale Atomenergie-Organisation (IAEO) 1972) unterzeichnet und ratifiziert. Darin werden die Rechte und Pflichten der einzelnen Vertragspartner spezifiziert, woraus unter anderem der Einsatz folgender Mittel durch die IAEO gerechtfertigt ist, um die Einhaltung des NVV zu verifizieren (siehe Internationale Atomenergie-Organisation (IAEO) 2001, S. 20):

- Auditierung der Materialbilanz des Betreibers und der Vergleich mit den an die IAEO deklarierten Bilanzen,
- Verifizierung des Spaltstoffinventars und dessen Änderungen durch Messen und Zählen,
- Anbringung und Überprüfung von Siegeln sowie Überwachungstechnik wie z.B. Kameras.

Diese Maßnahmen zielen in erster Linie darauf ab, die Richtigkeit der Deklarationen eines Staates zu überprüfen. Ob die Angaben auch vollständig sind, d.h. ob das gesamte spaltbare Material eines Staates gegenüber der IAEO deklariert wird, kann auf diese Weise hingegen nur begrenzt verifiziert werden. In der Geschichte des Nichtverbreitungsvertrages war dieser Ansatz daher nicht immer von Erfolg gekrönt.

Das wohl prominenteste Beispiel, an dem Unzulänglichkeiten traditioneller NVV-Verifikationsmethoden sichtbar wurden, ist die Geschichte der Einrichtung eines Kernwaffenprogramms im Irak durch die Regierung Saddam Husseins zwischen 1985 und 1991 (siehe Baute 2011). Als die Ausmaße dieses Programms sowie des Versagens der ausschließlichen Inspektion deklarerter Anlagen sichtbar wurden, verabschiedeten die Mitgliedstaaten der IAEO das Programm 93+2 (siehe Moyland 1997).

In diesem Programm werden Maßnahmen vorgeschlagen, welche die Vollständigkeit der Deklarationen eines Staates besser verifizieren können. Im Einzelnen wurden in diesem Zusammenhang folgende Instrumente und Methoden vorgeschlagen:

- unangekündigte Inspektionen,
- Inspektionen außerhalb deklarerter Anlagen,

- Analyse von Umweltproben beispielsweise von Wischproben aus einer Anlage,
- Verbesserung der technischen Mittel bei Siegeln und Überwachungstechnik,
- Verbesserung der Informationsanalyse,
- Nutzung zusätzlicher Informationen aus offenen Quellen sowie von anderen Mitgliedstaaten gelieferte Informationen.

Da ein Teil der genannten Maßnahmen durch das in INFCIRC/153 formulierte Mandat nicht abgedeckt ist, wurde der Abschluss des sogenannten Zusatzprotokolls nötig (siehe Internationale Atomenergie-Organisation (IAEO) 1997). Eine Ratifikation dieses Abkommens ist jedoch nicht obligatorisch und wird wegen der erweiterten Zugriffsrechte für die IAEO von einigen Mitgliedstaaten abgelehnt. Für die Nutzung zusätzlicher Informationen, was auch den Einsatz von Satellitenbildern im Rahmen der Arbeit der IAEO einschließt, ist eine Ratifikation des Zusatzprotokolls jedoch nicht notwendig.

Für die Überwachung des Nichtverbreitungsvertrages können Satellitenbilder prinzipiell für drei Aspekte Nutzen stiften. Erstens kann zumindest teilweise die Korrektheit der Deklarationen eines Staates verifiziert werden, indem die IAEO beispielsweise die deklarierten Ausmaße eines Gebäudes anhand von Satellitenbildern validiert. Zweitens ermöglichen Fernerkundungsaufnahmen die Entdeckung nicht deklarerter Aktivitäten wie etwa der Fall der Anreicherungsanlage im iranischen Fordo zeigte (siehe The White House Office of the Press Secretary 2009). Drittens sind Fernerkundungsdaten bei der Vorbereitung von Inspektionen von Nutzen, damit die Inspektoren gezielt an jenen Stellen einer Anlage eingesetzt werden können, an denen Fragen bzw. Unstimmigkeiten im Zusammenhang mit der Deklaration bestehen. Der Vergleich zweier Fernerkundungsaufnahmen ist insbesondere im Hinblick auf die beiden zuletzt genannten Anwendungsfälle von großer Bedeutung.

Bereits eine einzelne kerntechnische Anlage kann ein Gebiet von mehreren Quadratkilometern umfassen. Gleichzeitig steigt die Anzahl der von der IAEO weltweit überwachten Anlagen kontinuierlich<sup>1</sup> Das Budget der IAEO unterliegt allerdings starken Beschränkungen, weshalb Safeguardsmaßnahmen möglichst effektiv und kosteneffizient einzusetzen sind. Bei der Verwendung von Fernerkundungsdaten zur Detektion von Änderungen können automatisierte Verfahren einen Beitrag zu einer Steigerung der Effektivität und Effizienz liefern. Traditionell beruht die Analyse von Fernerkundungsdaten jedoch in erster Linie auf manuellen Analysen. Zwar wurde in der Vergangenheit eine Reihe von automatischen Verfahren zur Änderungsdetektion entwickelt, wie in Abschnitt 1.3 gezeigt wird. Jedoch weisen diese Verarbeitungsmethoden eine Reihe von Nachteilen auf. Ziel dieser Arbeit ist es daher, ein Verfahren zu entwickeln, das automatisiert und robust Änderungen in Fernerkundungsdaten detektieren hilft.

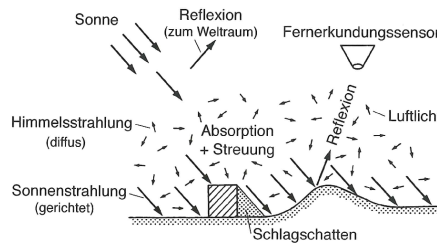
Der vorliegende Abschnitt soll dazu eine Einführung in die Grundlagen des Themas geben. Zunächst werden wesentliche Aspekte der Fernerkundung besprochen. Anschließend wird ein vielversprechendes Paradigma für die Bildauswertung von Fernerkundungsdaten, die objektbasierte Bildanalyse, im Hinblick auf die Anwendung für die Änderungsdetektion näher beleuchtet. In der Folge werden die Ziele der vorliegenden Arbeit im Detail definiert sowie die Struktur der Arbeit skizziert.

---

<sup>1</sup>Im Jahr 2010 standen 1170 kerntechnische Anlagen und Materialbilanzonen außerhalb kerntechnischer Anlagen unter IAEO-Safeguards (siehe Internationale Atomenergie-Organisation (IAEO) 2011). Im Jahr 2011 betrug diese Zahl 1212 (siehe Internationale Atomenergie-Organisation (IAEO) 2012) und im Jahr 2012 dann 1317 (siehe Internationale Atomenergie-Organisation (IAEO) 2013).

## 1.2. Fernerkundung

Unter dem Begriff Fernerkundung sollen im Rahmen dieser Arbeit angelehnt an die Definition in Albertz (2009) Verfahren zur Gewinnung von Informationen über die Erdoberfläche durch Messung und Interpretation der von ihr reflektierten oder emittierten elektromagnetischen Strahlung verstanden werden. Ein Modell der verschiedenen Arten von Strahlung ist in Abbildung 1.1 dargestellt.



**Abb. 1.1.:** Strahlungsmodell der Fernerkundung aus Albertz (2009).

Anwendung finden Daten der Fernerkundung in sehr vielen Bereichen. Historisch entwickelt hat sich die Fernerkundung aus der militärischen Aufklärung, wo sie auch heute noch große Bedeutung besitzt. Durch mehrere kommerzielle Satellitenmissionen sowie sinkende Kosten hat die Fernerkundung jedoch auch in zivilen Anwendungsbereichen Einzug gehalten. Neben vielen anderen Einsatzgebieten wird Fernerkundung in der Raumplanung, der Katastrophenhilfe, der Landwirtschaft sowie der internationalen Rüstungskontrolle verwendet.

Obwohl eine große Auswahl an möglichen Spektralbereichen für die Fernerkundung besteht, kommen für die vorliegende Arbeit ausschließlich die sichtbaren Wellenlängen des Lichts einschließlich des nahen Infrarotbereichs zum Einsatz. Motiviert wird diese Auswahl mit der einfacheren und direkteren Interpretierbarkeit der Fernerkundungsaufnahmen sowie der guten räumlichen Auflösung dieser Daten.

In den letzten Jahren hat die Sensorentwicklung von Erdbeobachtungssatelliten große Fortschritte gemacht. Insbesondere die räumliche Auflösung der Sensoren konnte stetig verbessert werden, so dass mittlerweile im zivilen Bereich der Satellitenfernerkundung Bodenauflösungen von 50 cm für Sensoren im Bereich des sichtbaren Lichts erreicht werden können. Für die Anwendung im Rahmen der NVV-Verifikation sind Bodenauflösungen besser als 1 m notwendig, um relevante Strukturen auf der Erdoberfläche detektieren zu können. Daten von Satellitenmissionen, deren Bodenauflösung in diesem Bereich liegt, werden auch als sehr hoch auflösende Satellitenbilder bezeichnet.

Ein weiterer wichtiger Aspekt der Fernerkundung für die NVV-Verifikation ist die kommerzielle Verfügbarkeit von Fernerkundungsdaten, um die Transparenz, Unabhängigkeit und Glaubwürdigkeit der IAEO sicherzustellen. Einige Satellitenmissionen, deren Daten diesen Anforderungen entsprechen, sind in Tabelle 1.1 aufgelistet.

Neben praktischen Erwägungen hat der Einsatz von Satelliten für die Erdbeobachtung auch rechtliche Aspekte. Der Weltraum ist laut Weltraumvertrag zur Erforschung und friedlichen Nutzung durch die gesamte Menschheit da. Andererseits berührt die Nutzung des Luftraumes die territoriale Souveränität eines Staates. Laut Hettling (2008, S. 21–23) ist jedoch die Nutzung eines Fernerkundungssatelliten zur Erfassung von Daten auf der Erdoberfläche anderer Staaten im Einklang mit dem Weltraumvertrag. Für die NVV-Verifikation verstößt somit die Nutzung von Fernerkundungsdaten nicht gegen internationales Recht. Darüber hinaus können Staaten ihrer Gesetzgebung unterliegenden Firmen jedoch Auflagen in Bezug auf die Erfas-

Name	Betreiber	Bodenauflösung	Wiederkehrzeit	Startdatum
Ikonos-2	GeoEye (USA)	1.0 m	ca. 3 d	09/1999
Quickbird-2	DigitalGlobe (USA)	0.63 m	3 - 7 d	10/2001
Kompsat-2	Kari (Südkorea)	1.0 m	ca. 3 d	07/2006
Worldview	DigitalGlobe (USA)	0.42 m	1.1 - 8.3 d	10/2008 & 10/2009
Pléiades	CNES (Frankreich)	0.7 m	min. 3 d	12/2011 & 12/2012

**Tab. 1.1.:** Satellitenmissionen mit sehr hochauflösenden optischen Sensoren, deren Daten kommerziell erhältlich sind, sortiert nach Startdatum.

sung und Verbreitung von Fernerkundungsdaten erteilen. Auch ist es gängige Praxis, dass der Export von Fernerkundungstechnologien an Bedingungen hinsichtlich ihres Einsatzes geknüpft wird. Ein Beispiel dafür ist die sogenannte *shutter control* der US-Regierung hinsichtlich Fernerkundungsdaten, die von Israel aufgenommen werden (siehe Williams 2011). Ebenso wichtig wie die Methoden der Gewinnung von Fernerkundungsinformation sind Techniken zur Analyse der Daten. Während in den Anfangszeiten der Fernerkundung die Aufnahmen in analoger Form vorlagen und visuell interpretiert wurden, hielt durch die Zunahme digitaler Daten auch die rechnergestützte Analyse im Rahmen der digitalen Bildverarbeitung Einzug. Durch die CCD-Technologie liegen die Daten in Form rechteckiger Raster vor. Ein Datenpunkt dieses Rasters wird als Pixel bezeichnet und bildet die Grundlage der Analyse. Die diskrete Form des Rasters sowie eine digitale Quantisierung der Pixelwerte ermöglichen eine automatisierte Interpretation der Daten mithilfe des Computers. Über die Analyse der einzelnen Pixel hinaus sind in der Vergangenheit auch Verfahren vorgeschlagen worden, die benachbarte Pixel zu Gruppen verbinden und diese als Objekte interpretieren. Diesen Ansatz nennt man objekt-basierte Bildanalyse. Im folgenden Abschnitt wird dieses neue Paradigma der Bildauswertung im Detail besprochen. Da dieser Abschnitt nur einen sehr knappen Überblick der Fernerkundung geben kann, sei für weiterführende Informationen auf die Literatur in Albertz (2009), Rees und Rees (2012) und Lillesand u. a. (2008) verwiesen.

### 1.3. Aktueller Forschungsstand

#### 1.3.1. Änderungsdetektion

Aus den mithilfe der Fernerkundung gewonnenen Daten können Informationen über Änderungen auf der Erdoberfläche abgeleitet werden. Dieser Prozess wird Änderungsdetektion genannt. Man versteht darunter nach Singh (1989) die Identifikation von Unterschieden im Zustand eines Objekts oder Phänomens durch seine Beobachtung zu verschiedenen Zeitpunkten. Die vorliegende Arbeit schränkt den Begriff weiterhin dahingehend ein, dass nur solche Veränderungen berücksichtigt werden sollen, die durch den Vergleich zweier durch Fernerkundungstechniken aufgenommener Bilder beobachtbar sind. Prinzipiell existieren nach Stow (2010) verschiedene Szenarien der Änderungsdetektion, die sich nach der Art der verwendeten Daten richtet. Zum einen gibt es das Szenario des Bild-zu-Bild-Vergleichs. Zweitens ist auch ein Bild-zu-Karte-Vergleich möglich. Schließlich ist auch ein hybrides Szenario denkbar, bei dem beispielsweise der Bild-zu-Bild-Vergleich durch Kartendaten unterstützt wird. Diese Arbeit konzentriert sich auf das Szenario des Bild-zu-Bild-Vergleichs. Als Anwendungsfälle für die Änderungsdetektion listet Stow (2010) allgemein die Analyse von Landnutzungs- bzw. Landbedeckungsänderungen, die Aktualisierung von Karten, die Verfolgung und Messung von Objektbewegungen und -verschiebungen auf der Erdoberfläche sowie die Detektion, Inventur und Kartierung von spezifischen Objekttypen auf der Erdoberfläche.

Insbesondere der zuletzt genannte Punkt ist für die vorliegende Arbeit von Interesse. Neben den allgemeinen Anwendungen, finden sich in Stow (2010) auch thematisch spezifische Verwendungsbereiche der Änderungsdetektion. In Anlehnung an die fünf Säulen des COPERNICUS-Forschungsprogramms (siehe *Copernicus - The European Earth Observation Programme* 2013) der EU sind dazu aufzuführen der Bereich der Forst- und Landwirtschaft, wo Änderungsdetektion mittels Fernerkundungsdaten unter anderem zur Abschätzung der Ernte sowie der Detektion von Anomalien und Krankheiten zum Einsatz kommt. In der Stadt- und Raumplanung kann mit Änderungsdetektion der Bevölkerungsdruck im zeitlichen Verlauf untersucht und beobachtet werden. Für das Umweltmonitoring bietet die Änderungsdetektion Möglichkeiten, unter anderem die Gletscherschrumpfung, Naturschutzgebiete, Desertifikation oder unerlaubten Ressourcenabbau zu überwachen. Im Rahmen des Katastrophenschutzes kann fernerkundungs-basierte Änderungsdetektion im Falle von Erdbeben, Überschwemmungen oder humanitären Krisen zum Einsatz kommen, um betroffene Gebiete zu identifizieren sowie die Planung der Aktivitäten zu erleichtern. Für den Anwendungsbereich Sicherheit ergeben sich durch die Nutzung von Änderungsdetektion Möglichkeiten des Einsatzes für die Überwachung von Truppenverschiebungen, Staatsgrenzen, Migrationsrouten sowie zur Unterstützung der Verifikation von Rüstungskontroll- und Abrüstungsverträgen. Da sich diese Arbeit im Kontext der NVV-Verifikation befindet, spielt der zuletzt genannte Anwendungsfall eine große Rolle.

Um das Problem der Änderungsdetektion zu lösen, sind viele verschiedene Herangehensweisen denkbar. In der Vergangenheit wurden diese Verfahren zumeist auf Grundlage der Pixel durchgeführt. Aus Sicht der Verfahren lassen sich vier Gruppen identifizieren. Die erste Gruppe, sogenannte differenzen-basierte Verfahren, bilden die Differenz zwischen den beiden Bildern und ermitteln auf dieser Grundlage beispielsweise über einen Schwellenwert wesentliche Änderungen zwischen den beiden Aufnahmezeitpunkten (siehe z.B. Bruzzone und Prieto 2000). In einer zweiten Gruppe befinden sich Ansätze, welche nicht direkt die Intensitäten oder Filterantworten zum Aufspüren der Änderungen nutzen, sondern zunächst den Merkmalsraum transformieren, so dass relevante Änderungen betont werden (siehe z.B. Nielsen u. a. 1998; Wiemker 1997). Die dritte Gruppe bilden sogenannte klassifikations-basierte Verfahren. Diese Methoden klassifizieren die beiden Bilder unabhängig voneinander und vergleichen anschließend die Klassifikationen miteinander, um zu einer Aussage über die Änderungen zwischen den Aufnahmezeitpunkten zu gelangen (siehe z.B. Coppin und Bauer 1996). Schließlich sind in einer vierten Gruppe all jene Verfahren versammelt, die Änderungssignaturen mithilfe eines Modells vorhersagen (siehe z.B. Kennedy u. a. 2007).

In der Literatur finden sich diverse Veröffentlichungen, die versuchen, einen Überblick der existierenden Verfahren zu geben. Ein früher Versuch den Forschungsstand festzuhalten findet sich in Singh (1989). Dabei werden die Verfahren gemäß des gewählten mathematischen Ansatzes bzw. wesentlicher Prozesscharakteristiken in Gruppen eingeteilt. Studien, welche die Ergebnisse verschiedener Verfahren zur Änderungsdetektion auf Basis eines einheitlichen Datensatzes vergleichen, finden sich unter anderem in Coppin und Bauer (1996) und Mas (1999). In Lu u. a. (2004) findet sich ein umfassender Überblick der Änderungsdetektionstechniken für Fernerkundungsanwendungen. Neben Vor- und Nachteilen der einzelnen Verfahren werden darin auch die Anwendungsgebiete beleuchtet sowie Empfehlungen für zukünftige Entwicklungen gegeben. Coppin u. a. (2004) betrachtet darüber hinaus auch Anforderungen an die Daten sowie an die Techniken zur Vorverarbeitung. Bei Aplin (2004) findet sich eine Übersicht der Sensoren, Techniken und Anwendungen der Änderungsdetektion im Bereich des Umweltmonitorings. Eine Betrachtung von Änderungsdetektionstechniken auch außerhalb der Fernerkundung findet sich in Radke u. a. (2005). Eine neuere Übersicht von Techniken, welche neben der Verarbeitung von optischen Daten auch Radar-Daten und 3D-Änderungsdetektion umfasst, findet sich Dekker u. a. (2009).

Alle genannten Verfahren zur Änderungsdetektion haben den Vorteil, dass sie im Gegensatz



zu einer durch menschliche Analysten durchgeführten Änderungsanalyse große Datenmengen in kurzer Zeit mit gleichbleibender Genauigkeit bearbeiten können. Andererseits entstehen durch die Automatisierung sowohl Fehler erster Art, d.h. es werden in der Realität nicht existente Änderungen als solche angezeigt, als auch Fehler zweiter Art, d.h. reale Änderungen werden nicht durch den Algorithmus detektiert. Die Ursachen für diese Fehler liegen erstens in Unterschieden der atmosphärischen Zusammensetzung zwischen den beiden Aufnahmezeitpunkten, so dass beispielsweise Objekte auf der Erdoberfläche unterschiedlich erscheinen, obwohl keine Änderung stattgefunden hat. Zweitens können nicht korrekt registrierte Bilder bzw. unterschiedliche Aufnahmewinkel zu Fehlern in der Änderungsdetektion führen. Eine dritte Fehlerursache können Unterschiede im Sonnenstand zwischen den beiden Aufnahmen sein. Dadurch kann einerseits der Schattenwurf von Objekten verändert werden. Andererseits können die Eigenschaften des reflektierten Licht durch unterschiedliche Sonnenstände und den dadurch modifizierten Weg des Lichts durch die Atmosphäre in beiden Aufnahmen voneinander abweichen. Die vierte Fehlerursache für die Ergebnisse der Änderungsdetektion stellen Unterschiede im jahreszeitlichen Erscheinungsbild von Pflanzen, der sogenannten Phänologie, dar<sup>2</sup>.

Um die oben genannten Fehlerarten zu vermeiden bzw. zu beheben, bieten sich prinzipiell drei Möglichkeiten an. Die erste Möglichkeit besteht durch die Wahl der Aufnahmebedingungen. So sollten beide Aufnahmen möglichst von demselben Sensor aufgenommen werden, denn obwohl viele hochauflösende Satelliten im selben Spektralbereich operieren, können durch unterschiedliche Sensoreigenschaften Verzerrungen auftreten. Bezüglich des Satelliten ist zudem anzumerken, dass es von Vorteil sein kann, falls möglich denselben Aufnahmewinkel und Orbit für beide Bilder zu verwenden, weil dadurch Fehler in der Koregistrierung der Bilder verringert werden. Um Phänologiefekte zu vermeiden empfiehlt es sich, die beiden Aufnahmen zur selben Jahreszeit durchzuführen. Dasselbe gilt in Bezug auf die Tageszeit, um Effekte durch den Sonnenstand zu vermeiden sowie gleichbleibende atmosphärische Bedingungen zu erhalten. Aufnahmen mit Schnee bzw. Wolken sind im Allgemeinen weniger gut für die Detektion anthropogener Änderungen geeignet und sollten daher vermieden werden.

Neben der generellen Vermeidung der Fehlerursachen kommen als zweite Möglichkeit Verfahren der Vorverarbeitung in Betracht. Im Rahmen dieser Vorverarbeitung stehen Verfahren bereit, um Unterschiede in der Radiometrie der beiden Bilder zu verringern. Geometrische Korrekturverfahren wie Koregistrierungs- und Orthorektifizierungsverfahren können dazu beitragen, dass Pixel an derselben Position in beiden Bildern auch denselben Bereich der Erdoberfläche abbilden. Bildfilterung bietet schließlich die Möglichkeit, Bildrauschen und andere Sensoreffekte zu beseitigen. In Abschnitt 2.2 werden diverse Möglichkeiten zur Vorverarbeitung näher beleuchtet.

Die dritte Möglichkeit, wie den Fehlerursachen der Änderungsdetektion begegnet werden kann, besteht in einem alternativen Paradigma der Bildverarbeitung, der objekt-basierten Bildanalyse (OBIA). Dies ist Bestandteil des folgenden Abschnitts.

### 1.3.2. Objekt-basierte Bildanalyse

Obwohl die Pixel eines optischen Fernerkundungsdatensatzes immer der Ausgangspunkt der Änderungsdetektion bilden werden, ist von der direkten Verwendung der Bildpunkte aus diversen Gründen abzuraten. Zum einen repräsentiert ein einzelnes Pixel durch die steigende räumliche Sensorauflösung einen immer kleiner werdenden Bereich auf der Erdoberfläche. Dadurch ist es möglich, Änderungen mit sehr geringer räumlicher Ausdehnung zu erfassen, aber gleichzeitig sinkt der Abstand zwischen Signal und Rauschen, die sogenannte *Signal-To-Noise Ratio* (SNR). Außerdem führen kleine Fehler in der Koregistrierung der Bilder bereits zu Fehlern, weil nicht die korrekten Pixel miteinander verglichen werden. Aus den genannten Gründen

<sup>2</sup>Die Auflistung der Fehlerursachen ist an die Arbeit in Chen u. a. (2012, S. 4436–4438) angelehnt.

geht die objekt-basierte Bildanalyse von einzelnen Pixeln über zur Aggregation benachbarter Pixel in homogenen Bereichen, den Objekten.

Castilla u. a. (2008) definiert die objekt-basierte Bildanalyse im Bereich der geographischen Informationssysteme (GIS) als automatisierte Methode, die einen räumlichen aufgelösten Datensatz zunächst in räumlich zusammenhängende Einheiten, sogenannte Objekte, zerlegt, die sich hinsichtlich der spektralen Eigenschaften homogen zeigen, jedoch von benachbarten Einheiten spektral abgrenzbar sind. Anschließend wertet die objekt-basierte Bildanalyse die Daten auf Grundlage der Objekteigenschaften in Hinblick auf räumliche, spektrale und temporale Aspekte aus, um dadurch geographische Information in einem GIS-kompatiblen Format zu generieren.

Da sich dieser Ansatz auf die speziellen Anforderungen der Anwendung im Bereich der Geoinformatik bezieht, spricht man in diesem Zusammenhang auch von *Geographic Object-based Image Analysis* (GEOBIA). Diese Anforderungen beinhalten insbesondere eine hohe Genauigkeit sowie die Möglichkeit Daten unterschiedlicher Maßstäbe als auch verschiedener Typen, etwa Radar-, LIDAR- oder optische Daten, verarbeiten zu können. Des weiteren kann GEOBIA durch die Unterschiedlichkeit der Objektklassen nicht im Rahmen eines modell-gestützten Ansatzes ablaufen. Für eine detaillierte Diskussion der Besonderheiten sei auf Marpu (2009, S. 17f.) verwiesen.

Indem ein Bild in zusammenhängende Bereiche zerlegt wird, versucht man den Prozess des menschlichen Sehens zu imitieren (siehe Marpu 2009, S. 14). Das automatische Zerlegen oder Partitionieren des Bildes wird Segmentierung genannt und bildet ein zentrales Element von OBIA. Dabei sollen zwar möglichst bedeutsame Objekte extrahiert werden, jedoch erhebt das Verfahren in dieser Verarbeitungsstufe noch keinen Anspruch darauf, geographische Objekte zu extrahieren. Die so ermittelten Segmente oder Objekte bilden die Grundlage der weiteren Analyse.

Die grundlegende Idee der räumlichen Aggregation von Pixeln zu Objekten wurde zwar bereits in relativ frühen Texten zur Fernerkundung durchgeführt, wie die Arbeit von Kettig und Landgrebe (1976) zeigt, in der ein Segmentierungsalgorithmus aus Rodd (1972) zum Einsatz kommt. Auch versuchten Cross u. a. (1988) mit einem *Split-Merge*-Segmentierungsalgorithmus und Ryherd und Woodcock (1996) mit Ansätzen zur Textursegmentierung, das Paradigma der pixel-basierten Bildanalyse in der Fernerkundung zu überwinden. Den eigentliche Beginn der GEOBIA-Entwicklung markiert jedoch die Präsentation des Bildverarbeitungssystems *eCognition* im Jahr 2000 (siehe Definiens Imaging, Munich 2009). Der darin verwendete Segmentierungsalgorithmus (siehe Baatz und Schäpe 2000) spielt auch heute noch eine große Rolle für GEOBIA-Anwendungen und markiert zudem einen wesentlichen Ausgangspunkt für diese Arbeit. GEOBIA hielt im Anschluss rasch Einzug in diverse Anwendungsbereiche der Fernerkundung wie z.B. die Raumplanung (siehe Meinel u. a. 2001), die Forstwirtschaft (De Kok 2001) oder auch die Verifikation des NVV (siehe Niemeyer 2001; Nussbaum und Menz 2008; Marpu 2009) um nur einige zu nennen. Ein sehr guter und umfassender Überblick der Einsatzgebiete von GEOBIA findet sich in Blaschke u. a. (2008). Für eine aktuelle Darstellung des GEOBIA-Forschungsstandes sei auf Blaschke (2010) verwiesen.

Neben den Methodenentwicklungen spielt auch die Einbettung der GEOBIA-Methoden in den Bildanalyseprozess der Fernerkundung eine große Rolle. Dazu ist einerseits die Entwicklung von Konzepten und theoretischen Fundierungen von GEOBIA wichtig. In diesem Zusammenhang sind besonders die Arbeiten von Schiewe (2002) und Lang und Blaschke (2006) interessant, weil darin Fragen nach der Natur der segmentierten Objekte sowie der Zusammenhang zwischen der Beobachtung in den Daten und der Ontologie im Kopf des Analysten gestellt werden. Andererseits kommt auch den Besonderheiten der Ergebnisevaluierung eine Bedeutung zu, da auch die Güte des neu hinzugefügten Segmentierungsschrittes begutachtet werden muss. Diesbezüglich sei besonders auf die Arbeiten von Neubert u. a. (2008) und Marpu u. a. (2010) verwiesen, in denen verschiedene Varianten der Segmentierungsgütebetrachtung vor-

gestellt und diskutiert werden.

Die Motivation für die Verwendung der objekt-basierten Methode begründet sich vor allem durch die damit verbundenen Vorzüge. Ein wesentlicher Vorteil liegt in der Verfügbarkeit von Objektmerkmalen, die über die spektralen Eigenschaften sowie den daraus berechneten Indizes wie etwa dem NDVI (siehe Marpu 2009, S. 28f.) bei der pixel-basierten Analyse hinausgehen. So sind beispielsweise für OBIA auch geometrische Eigenschaften der Objekte und Merkmale, die sich auf den Kontext eines Objekts in seiner räumlichen Nachbarschaft beziehen, in die Bildauswertung integrierbar. Für Textureigenschaften gilt zudem, dass sie im Gegensatz zur pixel-basierten Verarbeitung nicht auf einem beliebig geformten Filterfenster berechnet werden, sondern dass die Kalkulation für einen homogenen Bereich erfolgt. Dadurch werden verfälschende Effekte der Filter an den Objektgrenzen reduziert.

Weitere Vorteile resultieren direkt aus der Überwindung der Pixelstruktur. Pixel können geographische Objekte bezüglich ihrer Form nur bedingt abbilden, da ihre Konturen immer gleich sind und mit dem Untersuchungsgegenstand nicht in Verbindung stehen. Durch Segmentierung kann hingegen eine Form der Objektrepräsentation gefunden werden, welche die Realität besser beschreibt. Aus einem analogen Argument folgt, dass die Beziehungen der Objekte untereinander sowohl in der Bildebene als auch in einer möglichen Hierarchie durch den objekt-basierten Ansatz wesentlich besser modelliert werden können. Zwar existieren auch für pixel-basierte Ansätze sogenannte Laplace-Pyramiden (Sonka u. a. 2008, S. 109f.) zur Modellierung von Hierarchien, die sich jedoch bezüglich der Anpassung an die Bilddaten nicht so adaptiv zeigen wie ein objekt-basierter Ansatz.

Im Gegensatz zur pixel-basierten Methode kann mit GEOBIA zudem der Rechenaufwand gesenkt werden, da durch die Segmentierung die Anzahl der zu untersuchenden Objekte reduziert wird (Kettig und Landgrebe 1976). Dabei ist jedoch zu beachten, dass zum einen der Segmentierungsprozess selbst zusätzlichen Rechenaufwand erzeugt. Zum anderen steigt die Rechenzeit des Verfahrens durch den Einsatz aufwändig zu berechnender Objektmerkmale.

Schließlich liegt ein Vorzug des objekt-basierten Ansatzes in der bereits durch die oben genannte Definition begründeten Integrierbarkeit der Ergebnisse in Geoinformationssysteme. Während bei Ergebnissen in Form einzelner Pixel zunächst eine Polygonisierung stattfinden muss, kann das Resultat eines objekt-basierten Ansatzes direkt in ein GIS importiert werden. Naturgemäß bringen Vorteile eines Verfahrens auch zumindest einige Nachteile mit sich. GEOBIA bildet dabei keine Ausnahme. Beispielsweise wird in Hay und Castilla (2006) die Befürchtung geäußert, dass die Analysesoftware durch das neue Konzept zu kompliziert wird. Damit einher geht das bisher ungelöste Problem der Übertragbarkeit von Regelsätzen zur Analyse von einem Datensatz auf den nächsten. Die Software *eCognition* beispielsweise wurde konzipiert, die Analysetechniken eines menschlichen Analysten zu imitieren. Dafür benötigt die Software in der Umsetzung jedoch ein hohes Maß an Nutzerinteraktion, was zwar zu einer guten Ergebnisqualität jedoch auch zu sinkender Effizienz führt. Aus Sicht des Autors der vorliegenden Arbeit wurde bei GEOBIA bisher zu wenig Wert auf die Adaption pixel-basierter Methoden mit ihrem teilweise hohen Automatisierungsgrad gelegt.

Darüber hinaus führt die Segmentierung der Daten zu drei Problemen. Erstens gibt es laut Albrecht u. a. (2010) gut abgrenzbare, auch *bona fide* genannt, Objekte und gleichzeitig auch schlecht abgrenzbare, auch als *fiat* bezeichnete, Objekte. Beide Arten müssen jedoch mit demselben Segmentierungsalgorithmus extrahierbar sein. Dieses Problem kann laut Lang und Blaschke (2006) mit einem iterativen GEOBIA-Ansatz gelöst werden, was jedoch erneut zu vermehrter Nutzerinteraktion und somit zu einem geringeren Grad an Automatisierung führt. Zweitens konnte bisher in der Literatur noch nicht eindeutig geklärt werden, wann ein Objekt als korrekt extrahiert gilt. Zwar können durch Menschen erzeugte Referenzsegmentationen zu Hilfe genommen werden, jedoch kann dabei keine Eindeutigkeit erzielt werden. Drittens bedeutet die Segmentierung einen erheblichen zusätzlichen Rechenzeit- und Speicherbedarf. Schließlich muss auch die fehlende konzeptuelle Basis von GEOBIA als Nachteil angegeben

werden (siehe Hay und Castilla 2008). Daraus folgt, dass der Forschungsbereich in einzelne Anwendungsbereiche fragmentiert wird und die Arbeiten sich darauf konzentrieren, die Besonderheiten des eigenen Anwendungsbereichs hervorzuheben anstatt die Gemeinsamkeiten der GEOBIA-Methoden zu betonen.

Trotz der genannten Nachteile überwiegen die Vorzüge diese bei weitem. Die Überlegenheit objekt-basierter gegenüber pixel-basierten Verfahren konnte für die Anwendung der monotemporalen Bildklassifikation auf Grundlage von einzelnen Fallbeispielen unter anderem in Gao und Mas (2008), Chen u. a. (2008) und Perveen u. a. (2008) gezeigt werden. In Benz u. a. (2004), Hay und Castilla (2006) und Hay und Castilla (2008) finden sich die Grundlagen für die im Vorigen durchgeführte Diskussion der Vor- und Nachteile.

### 1.3.3. Objekt-basierte Änderungsdetektion

Wendet man das Paradigma der objekt-basierten Bildanalyse auf das Problem der Änderungsdetektion an, so gelangt man zu Verfahren der objekt-basierten Änderungsdetektion, dem zentralen Gegenstand der vorliegenden Arbeit. Im Folgenden soll ein Überblick in der Literatur vorhandenen Verfahren gegeben werden. Die Vor- und Nachteile des objekt-basierten Ansatzes für die Änderungsdetektion werden dabei ebenfalls beleuchtet.

Konzeptuell können laut Gamanya u. a. (2009) Objekte als physische oder gedachte Einheiten modelliert werden, die zudem identifizierbar sind. Erst diese von den sichtbaren Merkmalen unabhängige Objektidentität ermöglicht es in jedem Fall, verschiedene Objekte voneinander abzugrenzen. Das bedeutet auch, dass die Objektidentität nicht verloren geht, wenn sich die sichtbaren Eigenschaften eines Objekts ändern. Als Konsequenz dieser Betrachtung der objekt-basierten Änderungsdetektion werden in Raza und Kainz (2002), Blaschke (2005) sowie Gamanya u. a. (2009) folgende vier Gruppen von Formänderung eines Objekts identifiziert:

1. Erscheinen, Verschwinden und Kontinuität (Existenz);
2. Erweitern, Zusammenziehen und Deformieren (Größe und Form);
3. Versetzen und Rotieren (Position);
4. Fragmentieren und Amalgamieren.

Eine Fortführung dieses Konzepts findet sich im Ansatz der *Object Fate Analysis* (siehe Schöpfer und Lang 2006; Lang u. a. 2009).

Bezüglich der Algorithmen zur objekt-basierten Änderungsdetektion sind laut Chen u. a. (2012) vier verschiedene Verfahrenstypen denkbar. Im Folgenden wird jede der vier Gruppen kurz erläutert und auf einige Beispielstudien verwiesen.

Beim ersten Verfahrenstyp, dem Bildobjektvergleich, handelt es sich um Algorithmen, die Objekte zwischen den beiden Aufnahmezeitpunkten direkt anhand der Objektmerkmale vergleichen. Meist wird bei diesen Verfahren über einen Schwellenwert auf ein oder mehrere Merkmale die Grenze zwischen bedeutsamen und nicht bedeutsamen Änderungen definiert. Lefebvre u. a. (2008) beispielsweise nutzt zur Änderungsanalyse neben spektralen Eigenschaften auch Formeigenschaften der Objekte. Darüber hinaus sind auch transformierte Merkmale als Grundlage der Schwellenwertentscheidung verwendbar. Dies zeigen die Arbeiten in Niemeyer u. a. (2008) und Chen, Hutchinson u. a. (2007).

Die zweite Art von Verfahren, die klassifikations-basierten Methoden, zeichnet sich dadurch aus, dass beide Bilder unabhängig voneinander klassifiziert werden. Im Anschluss werden die Klassifikationen miteinander verglichen und auf diese Weise Änderungen identifiziert. Das hat den Vorteil, dass automatisch der Typ einer Änderung von einer Klasse in eine andere Klasse abgeleitet werden kann. Andererseits führt dieser Ansatz auch zu einer Fehlerfortpflanzung, da die Genauigkeit der Änderungsdetektion nur so hoch ist wie das Produkt

der Genauigkeiten der einzelnen Klassifikationen. Der klassifikations-basierte Ansatz findet Verwendung in Willhauck u. a. (2000), wo mithilfe von *Fuzzy-Membership*-Funktionen und *Nearest-Neighbour*-Klassifikatoren SPOT-Satellitendaten in Kombination mit Luftbildern auf Änderungen untersucht werden. Walter (2004) führt einen Vergleich zwischen einer Karte und einem Satellitenbild unter Einsatz der *Maximum-Likelihood*-Methode, wobei die Trainingsgebiete für die Klassifikation des Satellitenbildes der Karte entnommen werden, die als Grundlage des Vergleichs dient. Als abschließendes Beispiel für klassifikations-basierte Änderungsdetektion soll die Methode aus Kux und Araújo (2008) erwähnt werden, die auf einer regel-basierten Klassifikation der Einzelaufnahmen beruht.

Bei den Verfahren der dritten Kategorie handelt es sich um sogenannte multi-temporale Methoden, bei denen beide Bilder gleichzeitig der Änderungsanalyse als Eingabe dienen. Zwar wird die Komplexität bei diesem Ansatz im Vergleich zu klassifikations-basierten Verfahren erhöht, da neben räumlichen auch zeitliche Aspekte während der Analyse eine Rolle spielen. Andererseits hängt die Genauigkeit des Verfahrens nicht mehr vom Produkt der Genauigkeiten der einzelnen Klassifikationen ab. Beispiele für die Anwendung multi-temporaler Methoden finden sich beispielsweise in Desclée u. a. (2006), wo die unveränderten Objekte als multivariate Normalverteilung modelliert werden und mithilfe eines  $\chi^2$ -Tests Änderungen identifiziert werden. Hese und Schmulilius (2006) hingegen nutzt einen regel-basierten Ansatz auch unter Verwendung von Formeigenschaften, um Änderungen zu detektieren. In Conchedda u. a. (2008) schließlich kommt der *Nearest-Neighbour*-Algorithmus zum Einsatz. Zum Training des Algorithmus werden pro Änderungsklasse 25 Beispielobjekte eingesetzt.

In der vierten und letzten Gruppe von objekt-basierten Verfahren befinden sich sogenannte hybride Ansätze, die sowohl pixel- als auch objekt-basierte Ideen gleichzeitig verfolgen. Bei den meisten dieser Verfahren wird zunächst eine pixel-basierte Transformation des Merkmalsraumes durchgeführt, so dass die Änderungen zwischen den Aufnahmezeitpunkten in den Daten besser sichtbar werden. Anschließend wird ein objekt-basierter Ansatz für die weitere Verarbeitung, d.h. die Detektion und Klassifikation der Änderungen gewählt. Beispiele für diese Art von Verfahren finden sich in unter Verwendung IR-MAD-Transformation in Niemeyer u. a. (2005) sowie in McDermid u. a. (2008) mit der *Tasseled Cap Transformation*.

Die zuvor genannten vier Gruppen von Verfahren können nur einige Bereiche der objekt-basierten Änderungsdetektion abdecken. Für einen weiterführenden Überblick der vorhandenen Algorithmen sei auf Blaschke (2005), Stow (2010) sowie Chen u. a. (2012) verwiesen.

Als Vorzüge der objekt-basierten Änderungsdetektion gegenüber dem pixel-basierten Ansatz gelten zunächst einmal auch alle in Abschnitt 1.3.2 für GEOBIA genannten Punkte. Zusätzlich hat der objekt-basierte Ansatz im Hinblick auf die Koregistrierung einige bedeutsame Vorteile. Laut Desclée u. a. (2006) ist die objekt-basierte Änderungsdetektion weniger anfällig gegenüber Fehlern in der Koregistrierung. Man veranschaulicht sich dies leicht, wenn man den Effekt eines Positionsfehlers von der Größe eines Pixels betrachtet. Bei der pixel-basierten Bildanalyse wird das Ausgangspixel in der ersten Aufnahme mit dem benachbarten Pixel in der zweiten Aufnahme verglichen. Im Fall der objekt-basierten Bildanalyse und eines unveränderten Objekts hingegen bedeutet ein solcher Fehler lediglich die Verschiebung des Objekts um ein Pixel. Der Großteil der beiden Objektbeobachtungen überlappen sich somit und können je nach Methode weiterhin direkt miteinander verglichen werden. Zudem haben fälschlich erkannte Änderungsobjekte, die aus Koregistrierungsfehlern resultieren, laut Stow 2010 charakteristische geometrische Eigenschaften und können somit durch Klassifikation entfernt werden.

Allerdings existieren neben den Vorteilen der objekt-basierten Bildanalyse auch eine Reihe Probleme mit den bisher verfügbaren Verfahren. Im folgenden Abschnitt werden einige dieser Probleme aufgezeigt. Aus diesen Herausforderungen werden anschließend die Ziele der vorliegenden Arbeit abgeleitet.

## 1.4. Ziele der Arbeit

Im vorangegangenen Abschnitt wurde dargelegt, dass der Einsatz des objekt-basierten Ansatzes auch im Rahmen der Änderungsdetektion sinnvoll ist und bereits in mehreren Studien praktisch eingesetzt wurde. Die klassifikations-basierte Herangehensweise, ein häufig gewählter Ansatz zur objekt-basierten Änderungsdetektion, fußt jedoch auf expliziten Regeln und den diesen Regeln zugrundeliegenden Objekteigenschaften. Mit diesen Regeln, die speziell auf einen Datensatz zugeschnitten sind, wird versucht, Änderungen zu detektieren. Das führt zu einer erheblichen Reduktion im Automatisierungsgrad des Prozesses sowie der Notwendigkeit einer Adaption der Regeln, um andere Datensätze zu verarbeiten.

Dieser sogenannte *Image-driven Approach* wurde bereits in Marpu (2009) für GEOBIA im Allgemeinen kritisiert. Als Alternative wird eine Herangehensweise vorgeschlagen, die Methoden des maschinellen Lernens verwendet und auf diese Weise einen höheren Automatisierungsgrad erreicht. Diese Kritik am *Image-driven Approach* trifft auch auf den speziellen Fall der objekt-basierten Änderungsdetektion zu. **Daher soll die vorliegende Arbeit einen Beitrag zur Entwicklung eines objekt-basierten Verfahrens zur Änderungsdetektion liefern, das den Bedarf an Nutzerinteraktion für den Prozess so gering wie möglich hält.** Ausgangspunkt ist dafür der bereits erwähnte Ansatz zur objekt-basierten Änderungsdetektion aus Niemeyer u. a. (2008), wobei sich eine Weiterentwicklung des Verfahrens in John (2009) findet. Neben der Segmentierung beruht es im Wesentlichen auf der IR-MAD-Transformation mit anschließender automatischer Klassifikation der Änderungen.

Einen wichtigen Schritt bei der objekt-basierten Änderungsdetektion stellt die Segmentierung dar, da dies die Grundlage der weiteren Analyse bildet. In Niemeyer u. a. (2008) wurden drei prinzipielle Methoden der automatischen Segmentierung für die objekt-basierte Änderungsdetektion aufgezeigt: die bitemporale Segmentierung unter gleichzeitiger Verwendung beider Aufnahmen, die Segmentierung nur auf Grundlage einer der beiden Aufnahmen sowie die unabhängige Segmentierung beider Aufnahmen. Alle drei Ansätze haben jedoch wesentliche Nachteile. Unter anderem unterstützen die beiden zuerst genannten Verfahren den objekt-basierten Ansatz nur unvollständig, weil sie keine Möglichkeit der Verwendung von Formeigenschaften der Merkmale zulassen und nur unzureichend an die Daten adaptieren können. Beim dritten Verfahren fehlt es an Stabilität der Ergebnisse, da Änderungen zwischen den Segmentierungen auch in Bereichen auftreten, wo keine relevanten Änderungen zwischen den Aufnahmezeitpunkten stattgefunden haben. Eine detaillierte Betrachtung der drei Segmentierungsvarianten findet in Abschnitt 2.3.4 statt.

Um diese Unzulänglichkeiten der Segmentierung für die objekt-basierte Änderungsdetektion zu beheben, wird in Listner (2008) ein stabiler Segmentierungsalgorithmus in Anlehnung an die Methode aus Baatz und Schäpe (2000) vorgeschlagen. In Listner (2008) wird ausschließlich die Durchführbarkeit des Verfahrens durch eine Implementierung in *MATLAB* (siehe Mathworks 2005) gezeigt. Zudem bleiben einige wichtige Fragen zu Verfahrensdetails ungeklärt. Die vorliegende Arbeit soll daher den Segmentierungsalgorithmus für die objekt-basierte Änderungsdetektion aus Listner (2008) zum einen methodisch erweitern. Zum anderen soll der implementierte Algorithmus in die objekt-basierte Bildverarbeitungssoftware *eCognition* sowie den Ablauf der Änderungsdetektion allgemein integriert werden. Eine Optimierung des Algorithmus hinsichtlich Laufzeit und Speicherbedarf ist dabei wünschenswert. Für den Einsatz in der Praxis wird zudem eine Hilfestellung bei der Wahl der Parameter sowie die Unterstützung zur Verarbeitung großer Datensätze benötigt.

Im Verlauf des Prozesses der objekt-basierten Änderungsdetektion werden Paare von Objekten bzw. deren Merkmale miteinander verglichen. Im Fall der Verwendung von verschiedenen Segmentationen für die beiden Aufnahmen kann die Eindeutigkeit der Paarzuordnung jedoch nicht garantiert werden. In der vorliegenden Arbeit sollen daher Methoden zur Verknüpfung zweier Segmentationen sowie die Berechnung der Merkmalspaare bei Nichteindeutigkeit der

Zuordnung entwickelt werden, damit die Integration der Segmentierung in den Gesamtprozess sichergestellt werden kann. Darüber hinaus soll bei der Merkmalsauswahl ein Ansatz entwickelt werden, wie aus der großen Anzahl verfügbarer Objekteigenschaften eine Auswahl getroffen werden kann.

Für die Detektion von Änderungen ist die Transformation des Merkmalsraumes von großer Bedeutung, damit relevante Änderungen lokalisiert werden können. Die Arbeiten in Niemeyer u. a. (2008) und John (2009) verwenden dazu den IR-MAD-Algorithmus. Eine als Plugin für *eCognition* integrierte Version dieses Verfahrens liegt im Rahmen dieser Arbeit vor. In einer Vorstudie zur vorliegenden Arbeit wurden diverse Probleme mit dieser Implementierung festgestellt hinsichtlich numerischen Verhaltens festgestellt. Diese Fehler sollen im Rahmen dieser Arbeit behoben werden. Zudem soll das Plugin erweitert werden, so dass die Parameter der Transformation gespeichert werden und somit die Verarbeitung großer Bilder im *Tiling-Stitching*-Verfahren erfolgen kann.

Das Auffinden relevanter Änderungen bzw. die Gruppierung der Änderungen nach ihrer jeweiligen Änderungsart erfolgt in John (2009) mithilfe des FMLE-Algorithmus, eines unüberwachten Klassifikationsverfahrens, das ausschließlich die Anzahl der zu verwendenden Klassen als Parameter benötigt. In der Anwendung muss dieser Parameter jedoch trotzdem gewählt werden, wofür in der vorliegenden Arbeit eine Lösung gesucht werden soll. Außerdem kommt in John (2009) ein Schwellenwert zum Einsatz, der vor der Anwendung des FMLE-Algorithmus die Änderungen nach ihrer Intensität anhand eines Objektmerkmals aus der IR-MAD-Transformation vorfiltert. Die Bestimmung des Schwellenwertes kann mit dem SEaTH-Verfahren aus Nussbaum u. a. (2006) erfolgen. Dabei wird implizit die Normalverteilung des Merkmals angenommen. Da diese Annahme nicht in jedem Fall gerechtfertigt erscheint, soll im Rahmen der vorliegenden Arbeit die SEaTH-Methode auf andere Verteilungen verallgemeinert werden.

Über die zu entwickelnden Verfahren hinaus hat die vorliegende Arbeit zum Ziel, die praktische Anwendbarkeit der Methode zur objekt-basierten Änderungsdetektion zu zeigen. Dazu soll ein umfassender Test der Verfahren anhand simulierter und realer Daten erfolgen.

Nicht Teil der vorliegenden Arbeit ist die Entwicklung von Verfahren zur Vor- und Nachverarbeitung etwa der Koregistrierung der Bilder. In diesem Bereich wird auf bereits existierende Verfahren zurückgegriffen. Auch in Bezug auf Textureigenschaften von Objekten kann diese Arbeit keine Neuentwicklung leisten, verweist jedoch an den entsprechenden Stellen auf relevante Literatur. Der Anwendungsteil dieser Arbeit schließlich soll in erster Linie die Funktionsweise und die praktische Umsetzung demonstrieren. Eine tiefgreifende Studie des jeweiligen konkreten Anwendungsfalls bildet hingegen keinen Schwerpunkt dieser Arbeit.

## 1.5. Struktur der Arbeit

Die weitere Arbeit gliedert sich im Wesentlichen in drei Teile. Das Kapitel 2 bietet zunächst die umfassende Darstellung eines Verfahrens zur objekt-basierten Änderungsdetektion einschließlich der Betrachtung des Forschungsstandes für dessen Teilschritte. Illustriert werden die Verfahrensschritte anhand simulierter Bilddaten. Nicht alle Bestandteile der Methode wurden im Rahmen der vorliegenden Arbeit neu entwickelt. Eine Auflistung der Abschnitte mit methodischen Neuerungen soll jedoch im Folgenden gegeben werden.

- Entwicklung und Implementierung eines Segmentierungsalgorithmus für die objekt-basierte Änderungsdetektion in Unterabschnitt 2.3.4,
- Entwicklung und Implementierung einer Methode zur Segmentierung großer Bilder in Unterabschnitt 2.3.5,

- Entwicklung und Implementierung einer Methode zur Verknüpfung unterschiedlicher Segmentationen in Unterabschnitt 2.4.2,
- Entwicklung einer Methode zur Merkmalsauswahl in Unterabschnitt 2.4.3,
- Entwicklung von Methoden zur Transformation des Merkmalsraumes in den Unterabschnitten 2.5.3 und 2.5.4,
- Erweiterung einer Methode zur schwellenwert-basierten Klassifikation in Unterabschnitt 2.6.2,
- Anpassung und Erweiterung eines Algorithmus zur unüberwachten Klassifikation in Unterabschnitt 2.6.3.

In Kapitel 3 wird die Methode anschließend anhand von zwei Fallbeispielen erprobt und die Ergebnisse diskutiert. Das abschließende Kapitel 4 gibt eine Zusammenfassung der Arbeit sowie einen Ausblick auf mögliche zukünftige Entwicklungen.

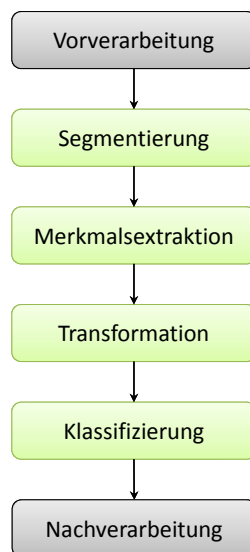




## 2. Methoden

### 2.1. Einleitung

Eine wesentliche Grundlage dieser Arbeit ist ein mehrstufiges Verfahren zur objekt-basierten Detektion von Änderungen zwischen zwei Bildern. Schematisch ist dieses Verfahren in Abbildung 2.1 dargestellt. Ausgangspunkt sind zwei digitale Multispektralaufnahmen desselben Gebietes aber unterschiedlicher Aufnahmezeitpunkte. Das Ziel des Prozesses ist die Herstellung einer Karte mit relevanten Unterschieden zwischen den Bildern. Betrachtet man die Aufnahmen in ihrem zeitlichen Kontext, so kann man von einer Karte von Änderungen sprechen.



**Abb. 2.1.:** Ablauf der objekt-basierten Änderungsdetektion.

In einem ersten Verarbeitungsschritt werden die beiden Aufnahmen geometrisch und radiometrisch vorverarbeitet. Einerseits ist das Ziel dieses Prozessschrittes, dass die beiden Bilder geometrisch zueinander passen, d.h. dass jedem Pixel im ersten Bild ein Pixel im zweiten Bild eindeutig zugeordnet ist, das zudem denselben Bereich auf der Erdoberfläche repräsentiert. Da die beiden gegebenen Bilder oft nicht von derselben Position aus mit derselben Aufnahme-richtung aufgenommen wurden, ist in den meisten Fällen eine geometrische Korrektur nötig. Auf Methoden, die dies bewerkstelligen, wird in Abschnitt 2.2 verwiesen.

Andererseits wird im ersten Verarbeitungsschritt auch eine radiometrische Korrektur durchgeführt. Deren Ziel ist es, Objekte mit gleichen Reflektionseigenschaften in beiden Bildern gleich darzustellen. Insbesondere sollten Bereiche ohne Veränderungen zwischen den Aufnahmezeitpunkten auch unverändert dargestellt werden. Durch Unterschiede im Sonnenstand und atmosphärische Effekte, wie sie z.B. durch Aerosole verursacht werden, können Bereichen un-

terschiedliche Bildwerte zugeordnet sein, obwohl keine Änderung an der Erdoberfläche stattgefunden hat. Auf Methoden, die radiometrische Korrekturen durchführen, wird in Abschnitt 2.2 verwiesen.

Der zweite Schritt des Prozesses ist die Segmentierung. Darunter versteht man die flächenhafte Extraktion von räumlich zusammenhängenden und möglichst homogenen Bildbereichen, Segmente oder Objekte genannt, aus den zwei korrigierten Bildern. Wie in Abschnitt 1.3 diskutiert, birgt die Weiterverarbeitung der Daten auf Basis von Objekten eine Reihe von Vorteilen gegenüber der Verarbeitung auf Grundlage der Pixel. Jedoch sind die meisten Segmentierungsalgorithmen wenig stabil gegenüber Änderungen in den Eingabedaten. Deshalb wird in Abschnitt 2.3.4 ein Verfahren vorgestellt, mit dessen Hilfe beide Bilder segmentiert werden können, so dass ihre Segmentierungen sich nur in den Bereichen unterscheiden, in denen auch wesentliche Änderungen an der Form der Objekte zu beobachten sind.

Bevor der eigentliche Vergleich auf Basis der segmentierten Bilder durchgeführt werden kann, ist es nötig, die Bilder bzw. ihre Segmentierungen auf geeignete Weise miteinander zu verknüpfen. Beim pixel-basierten Ansatz zieht man dazu einfach die Spektralwerte der Pixel an derselben Position heran. Im objekt-basierten Verfahren existiert jedoch einerseits nicht in allen Bildbereichen eine solche eindeutige Entsprechung zwischen den Objekten, wenn eine relevante Änderung der Form der Objekte und somit der Segmentierung festgestellt werden kann. Um eine Entsprechung zwischen den Objekten der beiden Szenen zu erzeugen, wird ein Verfahren in Abschnitt 2.4.2 vorgeschlagen. Andererseits liegt ein wesentlicher Vorteil der objekt-basierten Herangehensweise in der Nutzung von Eigenschaften der Textur, der Form und der Beziehung zwischen den Objekten untereinander. Aus dieser Vielzahl an Eigenschaften muss jedoch eine Auswahl getroffen werden. Dieses Problem wird in Abschnitt 2.4.3 diskutiert.

Zum Abschluss der Merkmalsextraktion liegen die Daten in Form von Paaren von Merkmalsvektoren vor. Im Vergleich der Vektorenpaare liegt der Kern der Änderungsdetektion. Denn ist der Abstand zwischen den Vektoren groß, so deutet dies auf eine Veränderung hin. Aufgrund dieser Betrachtung wäre die Differenzbildung der beiden Vektoren eine adäquate Herangehensweise. Große Differenzen würden dann auf relevante Änderungen hindeuten. Allerdings kann es sein, dass der Unterschied nicht nur lokal zwischen den beiden Vektoren besteht, sondern global aus einem systematischen Unterschied über das gesamte Aufnahmegebiet hinweg resultiert. Dann liegt es nahe, den Merkmalsraum so zu transformieren, dass relevante Unterschiede unabhängig von systematischen Einflüssen detektiert werden können. In Abschnitt 2.5 wird dazu unter anderem die Methode der Multivariaten Änderungsdetektion (MAD) auf den Einsatz für die objekt-basierte Änderungsdetektion angepasst. Die MAD beruht auf der kanonischen Korrelationsanalyse (KKA), einem klassischen Verfahren der multivariaten Statistik.

Durch die Transformation des Merkmalsraumes wird die Detektion von relevanten Änderungen einfacher. Jedoch müssen die Änderungen noch klassifiziert werden, um eindeutige Aussagen zu ermöglichen. Für diese Klassifikation eignen sich einerseits einfache schwellenwertbasierte Ansätze, falls nur eine Unterscheidung zwischen den Klassen „Änderung“ und „keine Änderung“ getroffen werden soll. Ist andererseits auch eine Differenzierung innerhalb der Klasse „Änderung“ nötig, so ist die Nutzung von komplexeren Methoden nötig. Dazu wird ein Verfahren der überwachten Klassifizierung vorgestellt, das anhand von nutzerdefinierten Trainingsgebieten ein neuronales Netz parametrisiert und mit dessen Hilfe den Rest der Szene in vorgegebene Klassen einordnet. Zwar liefert dieses Verfahren gute Ergebnisse. Jedoch müssen die Klassen und somit die möglichen Arten der Änderung vor dem Prozessschritt definiert werden. Das hat zur Folge, dass ein Änderungstyp, der nicht vom Nutzer erwartet wird, auch nicht korrekt klassifiziert werden kann. Um dieses Problem zu umgehen wird andererseits ein Verfahren aus der Familie der unüberwachten Klassifikationsmethoden vorgestellt, das als Eingabe nur die maximale Anzahl an Klassen benötigt. Um die optimale Klassenanzahl zu

bestimmen wird die Berechnung verschiedener Maßzahlen vorgeschlagen. Die angesprochenen Verfahren zur Änderungsklassifikation werden in Abschnitt 2.6 im Detail vorgestellt.

An dieser Stelle des Prozesses endet die eigentliche Änderungsdetektion und es folgt die Nachverarbeitung. Bisher besteht das Ergebnis aus einer Karte von Änderungen, die anzeigt, ob und evtl. was sich an einer Stelle der Szene geändert hat. Durch den objekt-basierten Ansatz ist diese Karte nicht nur für eine Rasterdarstellung geeignet, sondern die Objekte der Änderungen können auch als Polygone verstanden werden. Um die Datenmenge so gering wie möglich zu halten, sollte die Knotenanzahl der Änderungspolygone reduziert werden, wobei der Datenverlust dabei so klein wie möglich sein sollte. Anschließend empfiehlt sich der Export der generierten Informationen in ein geographisches Informationssystem, damit eine weitere Analyse ermöglicht wird. Diese Schritte werden in Abschnitt 2.7 erläutert. Außerdem muss die Güte der Analyse im Rahmen einer Fehlerbetrachtung beurteilt werden. Die dazu nötigen Konzepte und Methoden werden in Abschnitt 2.8 erläutert.

Das Verfahren der objekt-basierten Änderungsdetektion wurde im Rahmen dieser Arbeit nicht vollständig neu entwickelt, sondern basiert wesentlich auf den Arbeiten zur Multivariaten Änderungsdetektion in Nielsen u. a. (1998) sowie des grundlegenden Konzepts zur objekt-basierten Änderungsdetektion in Niemeyer u. a. (2008). Darüber hinaus werden im Folgenden jedoch wesentliche Neuerungen der Methoden vorgestellt, um einen vollständig objekt-basierten Änderungsdetektionsprozess implementieren zu können.

## 2.2. Vorverarbeitung

Zur Vorbereitung der Bilddaten auf die objekt-basierte Änderungsdetektion sind drei Schritte nötig: die geometrische Korrektur, die radiometrische Normierung sowie die Bildfilterung zur Entfernung von Rauschen. Auch wenn bei der vorliegenden Arbeit die Vorverarbeitung nicht im Fokus des Interesses steht, soll im Folgenden ein kurzer Überblick der dafür verfügbaren Methoden gegeben werden.

Der erste Vorverarbeitungsschritt, die geometrische Korrektur des Datensatzes, muss durchgeführt werden, um Störeinflüsse wie unterschiedliche Aufnahmeposition und -richtung der beiden Bilder zu korrigieren. Das konkrete Ziel ist dabei, dass sich zwei Pixel, die in beiden Bildern denselben Bereich auf der Erdoberfläche repräsentieren, auch an derselben Bildposition befinden. Um dies zu erreichen ist eine Registrierung der Bilder an einem gemeinsamen Koordinatensystem nötig. Falls beide Bilder mit ausreichender Genauigkeit georeferenziert sind, müssen sie nur in ein gemeinsames Raster gebracht werden. Alternativ kann eines der beiden Bilder in das Koordinatensystem des jeweils anderen transformiert werden. Um dies zu erreichen, können Verknüpfungspunkte, auch *Tie Points* genannt, verwendet werden. Diese Punktpaare markieren in beiden Bildern dieselbe Position auf der Erdoberfläche. Dadurch können die Parameter einer Transformationsfunktion ermittelt werden, welche die Koordinatensysteme der Bilder ineinander überführt. Als Verfahren eignen sich dazu Algorithmen der polynomialen Regression (siehe Niemeyer 2001, S. 26–29). Anschließend ist müssen die Werte zumindest eines der beiden Bilder neu abgetastet werden.

Mit dem zweiten Schritt der Vorverarbeitung, der radiometrischen Korrektur, wird beachtet, die spektralen Informationen beider Bilder von atmosphärischen Störeinflüssen zu befreien. Einerseits existieren dazu absolute Methoden wie die *Fast Line-Of-Sight Atmospheric Analysis of Spectral Hypercubes* (FLAASH) (siehe Matthew u. a. 2002), welche versuchen anhand von Informationen über den Zustand der Atmosphäre eine Korrektur der spektralen Information durchzuführen. Andererseits gibt es relative Methoden, auch radiometrische Normierung genannt, die auf der Idee beruhen, die Bilddaten so zu transformieren, dass Flächen mit ähnlichen spektralen Eigenschaften in beiden Bildern auch ähnlich repräsentiert werden. Dazu können die einzelnen Pixel als Punkte in einem Merkmalsraum betrachtet werden, der

durch die Kanäle der Bilder aufgespannt wird. Anhand dieses Modells kann kanalweise eine lineare Regression auf Grundlage der unveränderten Pixel der Bilder erfolgen. Eine robuste Schätzung der unveränderten Pixel erhält man z.B. mit dem IR-MAD-Algorithmus, der auch in Abschnitt 2.5.4 dieser Arbeit eine Rolle spielt. Für weitere Details zur radiometrischen Korrektur mithilfe des IR-MAD-Algorithmus sei auf die Arbeit in Canty und Nielsen (2008) verwiesen.

Im dritten Schritt der Vorverarbeitung, der sogenannten Bildfilterung, werden Störeinflüsse durch gegebenenfalls vorhandenes Sensorrauschen entfernt. Grundlage dieser Filter sind meist mathematische Faltungsoperationen, welche den Wert eines Pixels unter Einbeziehung der Pixelwerte seiner Umgebung neu bestimmen. Wenn etwa der Mittelwert der Pixelumgebung als neuer Pixelwert angenommen wird, so kann das Bildrauschen reduziert werden. Diese Art der Filterung nennt man Tiefpassfilter. Möchte man hingegen Kanten im Bild hervorheben, so eignen sich sogenannte Hochpassfilter. Komplexere Filter wie nichtlineare und morphologische Transformationen sind ebenfalls denkbar. Spezielle zur Vorbereitung des Segmentierungsschrittes in der objekt-basierten Bildanalyse wird in Marpu (2009) eine Filtermethode präsentiert, die versucht, die Heterogenität innerhalb der Objekte zu reduzieren und gleichzeitig die Unterschiede zwischen benachbarten zu vergrößern. Für einen Überblick der möglichen Techniken zur Bildfilterung sei auf Canty (2009, 107ff.) verwiesen. Bezüglich der Filterung ist jedoch anzumerken, dass diese Methoden sehr zurückhaltend angewandt werden sollten, da hierbei immer auch Bildinformationen verändert werden.

## 2.3. Segmentierung

### 2.3.1. Einleitung

Die objekt-basierte Bildverarbeitung unterscheidet sich von der traditionellen pixel-basierten Bildverarbeitung in erster Linie dadurch, dass nicht mehr einzelne Pixel, sondern Objekte Grundlage der Analyse sind. Objekte sind dabei räumlich zusammenhängende, möglichst homogene Pixelmengen. Den Prozess der Objektbildung aus den Pixeln nennt man Segmentierung. Das Ergebnis dieses Prozesses wird als Segmentation bezeichnet. Man verwendet für Objekte auch die synonyme Bezeichnung Segmente. Da das Gelingen der Segmentierung wesentlich für das Gesamtergebnis der objekt-basierten Bildanalyse und somit auch der objekt-basierten Änderungsdetektion ist, soll dieser Abschnitt das Problem und verschiedene Lösungsansätze näher beleuchten.

Eine Segmentation ist eine vollständige Aufteilung des Bildes in räumlich zusammenhängende, einander nicht überlappende, homogene Bereiche. Formal kann man eine Segmentation  $S$  als eine Partition des Bildes  $I$  in  $m$  Segmente  $s_1, \dots, s_m$  definieren. D.h.

$$\begin{aligned} I &= \bigcup_{i=1}^m s_i, \\ \forall i \neq j : s_i \cap s_j &= \emptyset, \\ \forall i : s_i &\neq \emptyset. \end{aligned} \tag{2.1}$$

Die Segmente sollen dabei räumlich zusammenhängend sein. Für jedes Paar von Pixeln  $(q, r)$  des Segments  $s$  muss also ein Weg bestehend aus einer Folge von Pixeln  $P = (p_1, \dots, p_k)$  innerhalb des Segments  $s$  existieren. Aufeinander folgende Pixel  $p_i$  und  $p_{i+1}$  müssen benachbarte Pixel sein<sup>1</sup>. Weiterhin müssen benachbarte Segmente heterogen sein, d.h. die Heterogenität innerhalb des Segments selbst soll möglichst klein sein. Alle Segmente  $s_i$  müssen dabei homogen

<sup>1</sup>An dieser Stelle sei bemerkt, dass diese Segmentdefinition auch der Festlegung eines Nachbarschaftskonzeptes bedarf. Im Konzept der sogenannten 4-Nachbarschaft werden jene Pixel die nördlich, östlich, südlich und

sein. Unter Verwendung eines Heterogenitätskriteriums  $H$  soll somit gelten

$$\begin{aligned} H(s_i \cup s_j) &= \text{wahr}, \forall i \neq j, i \text{ und } j \text{ benachbart,} \\ H(s_i) &= \text{falsch}, \forall i. \end{aligned} \quad (2.2)$$

In dieser Definition einer Segmentierung wird das Heterogenitätskriterium nicht näher spezifiziert und kann daher prinzipiell beliebig gewählt werden. Eine Methode bietet die Definition von Heterogenitätskriterien  $H(s)$  anhand von Bedingungen etwa Schwellenwerten an Heterogenitätsmaße  $h(s)$ . Von einfachen Heterogenitätsmaßen wie dem euklidischen Abstand im Spektralraum über Texturmaße bis hin zu komplexen semantischen Modellen der Objekte wurden eine Vielzahl von Heterogenitätsmaßen in der Fachliteratur vorgestellt. Allerdings entscheidet sich über die Wahl des Heterogenitätsmaßes und des zugehörigen Heterogenitätskriteriums auch die Qualität der Segmentation. Denn verwendet man z.B. ein Heterogenitätskriterium, das nur für identische Spektralwerte innerhalb eines Segments den Wert **wahr** annimmt, so können benachbarte Pixel, die sich nur durch Sensorrauschen voneinander unterscheiden, nicht in ein Segment gruppiert werden. Auch kann ein allein auf spektrale Informationen beruhendes Heterogenitätskriterium Segmente erzeugen, deren Form sehr „zerklüftet“ wirkt und der Intuition des menschlichen Betrachters widerspricht. Wiederum kann durch die Benutzung eines zu komplexen Heterogenitätskriteriums, das möglicherweise sogar nur auf einen Typ von Objekten zugeschnitten ist, nicht die gesamte Diversität des Bildes wiedergegeben werden, mit der Folge einer für die weitere Analyse unzureichenden Segmentation. Wie man sieht sind sehr viele Kriterien zur Beschreibung der Heterogenität eines Segments denkbar. Das ist auch ein Grund für die große Zahl an Segmentierungsverfahren, da in vielen Veröffentlichungen dieses Bereichs die hauptsächliche Neuerung in der Präsentation eines neuen Heterogenitätskriteriums besteht. Die Menge der verwendeten Algorithmen ist jedoch sehr begrenzt und oft Methoden aus dem Bereich der Optimierung entlehnt. Ist nämlich das Heterogenitätskriterium definiert, so lässt sich die Segmentierung auch als Optimierungsproblem begreifen, bei dem die durchschnittliche Objektgröße  $\bar{A}$ , definiert durch

$$\bar{A} := \frac{|I|}{|S|}, \quad (2.3)$$

maximiert werden soll unter der Nebenbedingung, dass die Segmente homogen sind. Äquivalent dazu kann auch die Objektanzahl  $|S|$  minimiert werden, da die Bildgröße  $|I|$  während der Segmentierung konstant ist. Ein Überblick über verschiedene Heterogenitätskriterien und Segmentierungsalgorithmen für Einzelbilder findet sich in Abschnitt 2.3.2.

Die Sicht auf die Segmentierung als Optimierungsproblem ist jedoch nur ein Modell der Realität. Selbst wenn ein globales Optimum gefunden wird, bedeutet dies nicht automatisch, dass die Segmentation selbst von guter Qualität ist. Daher stellt sich die Frage, wann eine Segmentierung als gut angesehen werden kann. Prinzipiell können dazu automatisierte Verfahren verwendet werden, die eine Bewertung der Segmentation vornehmen. Eine solche unüberwachte Quantifizierung der Güte einer Segmentation löst das Problem allerdings nicht endgültig, da ähnlich wie bei der Wahl des Heterogenitätskriteriums ein Modell definiert wird. Ein Ausweg ist eine überwachte Evaluierung der Segmentation, bei der ein Nutzer Referenzsegmente definiert und diese mit den generierten Segmenten in Bezug auf ihre Übereinstimmung verglichen werden. Mit der Frage der Evaluierung von Segmentationen beschäftigt sich ebenfalls der Abschnitt 2.8.2.

Die Ausführungen zu Segmentierungsverfahren bezogen sich bisher auf allgemeine Anwendungsfälle. Da insbesondere auch die große Anzahl der Veröffentlichungen mit neuen Segmentierungsalgorithmen angesprochen wurde, stellt sich die Frage, warum für die Bildverarbeitung in der Fernerkundung und insbesondere für die Änderungsdetektion nicht einfach ein

---

westlich eines Pixels liegen als dessen Nachbarn betrachtet. Bei dem Konzept der 8-Nachbarschaft werden zusätzlich noch jene Pixel nord-östlich, süd-östlich, süd-westlich und nord-westlich als Nachbarn angesehen.

bestehender Algorithmus aus einem anderen Bereich benutzt werden kann. In Schiewe (2002) werden fünf Aspekte genannt, worin sich die Segmentierung für Fernerkundungsdaten von anderen Anwendungsbereichen unterscheidet.

1. Durch zusätzliche Informationen wie Höhendaten und Informationen aus GI Systemen wird die Heterogenität der Daten und somit die **Komplexität der Eingabe** erhöht.
2. Fernerkundungsdaten sind oft multispektral, was neben der Komplexität auch die **Redundanz** erhöht.
3. Für die Analyse einer Szene sind **verschiedene Objekttypen** von Interesse. Ein Segmentierungsalgorithmus muss in der Lage sein, diese mit konstanter Qualität zu extrahieren.
4. Das zugrundeliegende Objektmodell aus anderen Anwendungsbereichen, etwa das Modell eines Herzens in der medizinischen Bildverarbeitung oder jenes eines Bordsteins in der Bildverarbeitung für automatische Fahrsysteme, ist nicht auf die Fernerkundung übertragbar bzw. ergäbe sich in der Anwendung für die Fernerkundung ein sehr **komplexes Objektmodell**.
5. Die **verschiedenen Skalen**, die in einer Fernerkundungsszene vorhandenen sind, wurden bisher noch nicht vollständig in der Analyse ausgenutzt.

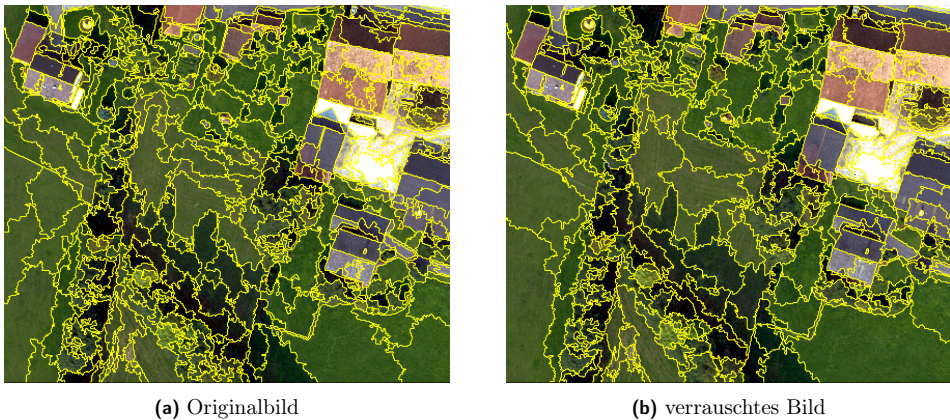
Insbesondere das in Punkt 5 erwähnte Problem der Multiskaligkeit wurde zwar seit der Veröffentlichung dieser Liste in diversen Publikationen behandelt (siehe z.B. Burnett und Blaschke 2003; Hay und Castilla 2008; Esch u. a. 2008). Für die nicht auf Klassifikationsmethoden basierende objekt-basierte Änderungsdetektion ist jedoch bis heute kein Verfahren bekannt, das die Multiskaligkeit der Daten vollständig in die Analyse integriert.

Aus den oben genannten Besonderheiten ergeben sich nach Schiewe (2002) drei Anforderungen an einen Segmentierungsalgorithmus, der im Bereich der Bildauswertung von Fernerkundungsdaten eingesetzt werden soll. Erstens sollte der Algorithmus unterschiedliche Eingabedaten gleichzeitig verarbeiten können (*multi source*). Zweitens muss der Algorithmus in der Lage sein, alle erwarteten Objekttypen korrekt zu extrahieren (*multi method*). Drittens ist es nötig, dass der Algorithmus mehrere Generalisierungsebenen erzeugt, da unterschiedliche Objekttypen eventuell unterschiedliche Maßstäbe zur geeigneten Darstellung benötigen (*multi scale*). Ausgehend von diesen Anforderungen hat sich in mehreren Studien (siehe Neubert u. a. 2008; Marpu u. a. 2010) herausgestellt, dass der *Multiresolution-Segmentation*-Algorithmus (MRS) aus Baatz und Schäpe (2000) sehr gut für den Einsatz im Bereich der Analyse von Fernerkundungsdaten geeignet ist. Da dieser Algorithmus grundlegend für diese Arbeit sind, erfolgt in Abschnitt 2.3.3 eine detaillierte Beschreibung.

Das Problem der Segmentierung ist laut Donoser u. a. (2011) und Hay und Castilla (2008) ein schlecht gestelltes Problem<sup>2</sup>, da das Ergebnis nicht stetig von den Eingangsdaten abhängt. Das hat zur Folge, dass ein Segmentierungsalgorithmus auch dann stark unterschiedliche Resultate produzieren kann, wenn sich die Eingabedaten nur leicht ändern. Für den *Multiresolution-Segmentation*-Algorithmus wird dieser Effekt beispielhaft in Abbildung 2.2 dargestellt.

Die nicht stetige Abhängigkeit der Lösung des Segmentierungsproblems von den Daten hat für die objekt-basierte Änderungsdetektion weitreichende Auswirkungen. Wenn nämlich die beiden zu vergleichenden Bilder mit demselben Segmentierungsalgorithmus und denselben Parametern unabhängig voneinander segmentiert werden und die Aufnahmen sich nur lokal unterscheiden, entstehen trotzdem Unterschiede in der Segmentierung in Regionen, in denen

<sup>2</sup>Ein gut gestelltes Problem zeichnet sich durch die Existenz und Eindeutigkeit seiner Lösung, sowie der stetigen Abhängigkeit der Lösung von den Eingabedaten aus. Ist mindestens eine der drei Bedingungen nicht erfüllt, so nennt man das Problem schlecht gestellt.



**Abb. 2.2.:** Ausgabe der Anwendung des *Multiresolution-Segmentation*-Algorithmus auf zwei Bilder, die sich nur durch additives Gaußsches Rauschen ( $\sigma = 0.5$ ) unterscheiden.

sich nichts geändert hat. Die Gründe dafür liegen laut Nussbaum und Menz (2008, S. 25) in Unterschieden der Radiometrie, der Beleuchtung und des Kamerawinkels. Hinzu kommen Änderungen der Phänologie und Fehler in der Koregistrierung, die das Segmentierungsergebnis beeinflussen. Zwar ist es möglich, diese Störfaktoren mit den in 2.2 beschriebenen Methoden teilweise zu korrigieren. Jedoch können in der Praxis identische Segmentierungen eines Gebietes nur bei identischen Bildern erreicht werden. Eine Diskussion dieses Aspekts für das Verfahren der *Multiresolution Segmentation* wird in Abschnitt 2.3.2 geführt. Zudem wird in Abschnitt 2.3.4 ein Verfahren vorgestellt, mit dessen Hilfe Fernerkundungsaufnahmen im Rahmen der objekt-basierten Änderungsdetektion so segmentiert werden können, dass nur Formänderungen zwischen den Bildern auch zu Änderungen in der Segmentierung führen.

Neben den konzeptionellen Aspekten der Segmentierung kommt es in der Praxis vor, dass durch die hohen Datenmengen in der Fernerkundung die Laufzeit der Segmentierungsalgorithmen sehr groß wird bzw. der Speicherbedarf stark wächst. Aus diesem Grund widmet sich der Abschnitt 2.3.5 dem Problem der Segmentierung großer Bilder. Eine naheliegende Herangehensweise an dieses Problem ist die Aufteilung eines großen Bildes in mehrere kleinere, die dann parallel verarbeitet und anschließend wieder zusammengefügt werden. Durch diese Aufteilung kommt es allerdings zu Artefakten an den Rändern der verarbeiteten Teilaufnahmen. Daher wird ein Verfahren zur Bereinigung vorgestellt, das eine Segmentation erzeugt, welche im Idealfall keine Artefakte mehr beinhaltet.

### 2.3.2. Verfahren zur Segmentierung von Einzelbildern

Um Segmentierungsverfahren für die objekt-basierte Änderungsdetektion zu entwickeln, ist es angeraten, zunächst einmal einen Blick auf die bestehenden Verfahren zur Segmentierung einzelner Bilder zu werfen. Es existieren sehr viele Anwendungsbereiche für die digitale Bildverarbeitung und folglich auch für die objekt-basierte Bildanalyse. Segmentierungsalgorithmen für Einzelbilder wurden im Kontext dieser verschiedenen Einsatzgebiete entwickelt und dabei oft genau auf deren Anforderungen zugeschnitten. Unter anderem stammen Segmentierungsverfahren aus den Bereichen medizinische Bildverarbeitung, Gesichtserkennung, autonome Fahrsysteme, materialwissenschaftliche Anwendungen und Fernerkundung. Zwar interessiert im Rahmen dieser Arbeit hauptsächlich der zuletzt genannte Anwendungsbereich, jedoch soll in diesem Abschnitt ein allgemeiner Überblick der Verfahren gegeben werden. Darüber hinaus finden sich Übersichten zu Verfahren der Bildsegmentierung auch in Haralick und Shapiro



(1985), Skarbek und Koschan (1994), Niemeyer (2001, S. 128), Cheng u. a. (2001) und Sonka u. a. (2008, S. 175–320).

Die Verfahren zur Segmentierung eines Bildes können entsprechend ihrer Herangehensweise an das Problem in folgende Kategorien eingeordnet werden:

1. **Schwellenwert-basierte Verfahren** teilen den Grauwertbereich eines Bildes und somit die Pixel in üblicherweise zwei Bereiche, die Objekte und den Hintergrund, ein. Benachbarte Pixel, die dem selben Bereich zugeordnet wurden, werden dann auch mit demselben Segment assoziiert. Die Bestimmung des Schwellenwertes ist mithilfe einer Vielzahl manueller sowie automatischer Ansätze und mathematischer Verfahren auf Basis der Grauwertverteilung möglich. Die Vorzüge schwellenwert-basierter Segmentierungsverfahren liegen in ihrer Einfachheit sowie der Flexibilität bezüglich des mathematischen Modells. Ein großer Nachteil dieser Ansätze ist die zugrundeliegende Annahme, dass die Objekte sich im gesamten Bild auf ähnliche Art und Weise vom Hintergrund unterscheiden, was bei ungleichmäßiger Beleuchtung keinesfalls gegeben sein muss. Zwar kann dieser Effekt etwa durch Bildfilterung oder durch lokale Schwellenwerttechniken gemildert werden, jedoch stoßen diese Ansätze bei komplexen Bildern rasch an ihre Grenzen. Für den Einsatz in der Fernerkundung sind sie daher weniger gut geeignet. Anwendung finden diese Algorithmen jedoch in Bereichen, in denen die Beleuchtungsbedingungen beeinflussbar bzw. relativ leicht zu modellieren sind, etwa bei der Schrifterkennung (siehe Leedham u. a. 2002; Leedham u. a. 2003) sowie im Bereich der zerstörungsfreien Prüfverfahren (siehe Sezgin und Sankur 2001). Eine allgemeine Übersicht zu schwellenwert-basierten Segmentierungsalgorithmen findet sich in Sahoo u. a. (1988).
2. **Kanten-basierte Verfahren** nutzen die Tatsache, dass sich an den Objektgrenzen die Grauwerte mitunter abrupt ändern. Durch die Nutzung sogenannter Kantenfilter, wie etwa dem Canny- oder dem Laplacefilter, können diese Änderungen im Bild hervorgehoben werden. Anschließend ist es unter Einbeziehungen des Bildkontextes möglich, relevante Kanten im Bild zu ermitteln. Eine Möglichkeit ist es, das Gradientenbild als topographisches Relief zu betrachten und darauf die sogenannte Wasserscheidentransformation (siehe Meyer 1994) anzuwenden. Die Idee dahinter ist die Betrachtung der Pixelwerte als Höheninformation. Zwei Pixel gehören in dieser Betrachtung zum selben Segment, wenn ein Wassertropfen von beiden Startpunkten zur selben Senke fließen würde. Wasserscheiden ergeben sich dann als Linien zwischen benachbarten Pixeln, bei denen ein Wassertropfen zu unterschiedlichen Senken fließen würde. Eine andere Variante, die Kanteninformation zu nutzen, ist der Einsatz von Algorithmen u.a. aus den Bereichen der Graphentheorie, der dynamischen Programmierung oder Relaxationsmethoden, um die Segmente zu extrahieren. Dabei kann jedoch das Problem auftreten, dass die Kantenzüge nicht geschlossen sind, was mit Methoden, wie sie etwa in Listner (2008, S. 20–21) beschrieben sind, gelöst werden kann. Auf Grundlage der Segmentgrenzen kann abschließend relativ einfach die Segmentierung des Bildes erzeugt werden. Für eine weiterführende Betrachtung der beschriebenen Ideen sei an dieser Stelle auf die Ausführungen in Sonka u. a. (2008, S. 184–222) und Nischwitz und Haberäcker (2004, S. 469–526) verwiesen. Obwohl der Ablauf kanten-basierter Segmentierungsverfahren auf den ersten Blick sehr komplex wirken mag, so ist der Rechenaufwand der Verfahren gering, weshalb sie sich effizient implementieren lassen. Es ist jedoch anzumerken, dass diese Verfahren oft die Darstellung der Objekte vor einem Hintergrund erwarten, was in den Anwendungen der Fernerkundung keine realistische Modellannahme ist. Weiterhin verstärken Kantenfilter das insbesondere in Fernerkundungsdaten schlecht zu unterdrückende Bildrauschen und führen somit zur Detektion von Kanten, wo eigentlich keine sind. Das hat eine Übersegmentierung des Bildes zur Folge. Schließlich sind Kantenfilter auf einen Grauwertkanal im Bild beschränkt und können daher die Multispektralität der

Daten nur indirekt Nutzen. Aus den genannten Gründen, werden wir kanten-basierte Segmentierungsverfahren im Rahmen dieser Arbeit nicht weiter betrachten.

3. Statt die Segmente aus ihren Begrenzungslinien zu konstruieren, erzeugen **regionen-basierte Verfahren** die Segmente direkt auf Basis ihrer inneren Heterogenität. Man unterteilt die regionen-basierten Segmentierungsverfahren in drei Klassen. Die erste Klasse, die *Split*-Verfahren, beginnen mit einem Segment, welches das ganze Bild repräsentiert, und teilen iterativ die Segmente auf, bis sie ausreichend homogen sind. Die zweite Klasse, die *Merge*-Verfahren, werden mit einer sogenannten Schachbrettsegmentierung initialisiert, in der jedes Pixel einem Objekt entspricht. Anschließend verbinden diese Algorithmen iterativ benachbarte Segmente solange die resultierenden Segmente homogen sind. Verfahren der dritten Klasse, die *Split-Merge*-Verfahren, kombinieren die beiden Ansätze und gelangen so zu einer Segmentierung des Ergebnisses. Ein wesentlicher Vorteil regionen-basierter Segmentierungsverfahren liegt in der algorithmischen Einfachheit und Robustheit des Verfahrens. Regionen-basierte Segmentierungsverfahren sind zudem sehr flexibel bezüglich sowohl des Heterogenitätskriteriums als auch des Optimierungsverfahrens. In der Tatsache, dass eine flächendeckende Segmentierung entsteht, besteht ein dritter großer Vorteil dieses Ansatzes. Auf der anderen Seite tendieren diese Verfahren dazu, Segmente ähnlicher Größe innerhalb des Bildes zu extrahieren. Auch resultieren verrauschte Bilder oft in einer Übersegmentierung der Szene, was jedoch durch die Nutzung geeigneter Filter oder Heterogenitätskriterien umgangen werden kann. Schließlich ist der Rechenaufwand bei regionen-basierten Verfahren oft sehr hoch. Jedoch wurde mit der Software *eCognition* gezeigt, dass trotzdem eine effektive Implementierung regionen-basierter Segmentierungsverfahren möglich ist. Trotz der genannten Nachteile überwiegen die Vorteile regionen-basierter Verfahren, weshalb im Rahmen dieser Arbeit der Fokus auf dieser Art der Segmentierung liegt. Ein Überblick regionen-basierter Segmentierungsverfahren findet sich in Sonka u. a. (2008, S. 223–237).
4. **Modell-basierte Verfahren** sind eigentlich Klassifikationsverfahren, die das Bild anhand eines Modells bzw. einer Vorlage nach Objekten einer bestimmten Klasse, z.B. Flugzeugen oder Schiffen, durchsuchen. Sie teilen das Bild ein in den Hintergrund und in Bereiche, in denen die Vorlage gefunden wurde. Durch diese Einteilung entsteht die eigentliche Segmentierung des Bildes. Wünschenswert ist es bei dieser Art der Segmentierung, dass die Verfahren invariant gegenüber Rotation, Translation und Skalierung des Musters sind. Eine weite Verbreitung finden die Verfahren im Bereich des Computer Vision (siehe Haralick und Shapiro 1992). Für die Analyse von Fernerkundungsdaten werden diese Verfahren zwar auch verwendet (siehe z.B. Li u. a. 2011). Da jedoch im Rahmen dieser Arbeit alle Arten anthropogener Änderungen detektiert werden sollen, birgt die Einschränkung, dass nur eine bestimmte Klasse von Objekten korrekt segmentiert werden kann, einen großen Nachteil. Aus diesem Grund werden modell-basierte Verfahren im Rahmen dieser Arbeit nicht weiter berücksichtigt. Für eine weiterführende Betrachtung modell-basierter Segmentierungsverfahren sei auf Sonka u. a. (2008, S. 237–241) verwiesen.

In Kombination mit Bildfiltern können die beschriebenen Ansätze auch dafür genutzt werden, texturierte Bereiche zu segmentieren. Unter Textureigenschaften seien in diesem Zusammenhang „Maße zur Beschreibung der Variation der Reflektionsintensität einer Oberfläche wie etwa Glattheit, Rauheit, Regelmäßigkeit“ (siehe Howe 1995) zu verstehen. Laut Haindl und Mikes (2008) existieren über eintausend verschiedene Algorithmen zur Textursegmentierung. Neben der großen Anzahl an Varianten haben diese Verfahren den Nachteil, dass spektral homogene Bereiche nicht immer korrekt segmentiert werden. Eine Lösung dieses Problems besteht in der Nutzung einer Kombination von spektralen Merkmalen und Textureigenschaften (sie-

he Ryherd und Woodcock 1996). Darüber hinaus setzt die Nutzung von Texturmerkmalen voraus, das ein Vorwissen über die Textur in einem Bild vorhanden ist. Eine ausführliche Diskussion des Aspekts Textur für die Segmentierung kann im Rahmen dieser Arbeit nicht gegeben werden. Jedoch finden Textureigenschaften neben den spektralen Eigenschaften als Eingabedaten der Segmentierung Verwendung. Dazu wird einerseits eine Bildfilterung mittels invarianter Momente eingesetzt (siehe Kristinsdóttir 2008). Andererseits werden die aus der Grauwertübergangsmatrix, auch als *Gray-Level Co-Occurrence Matrix* (GLCM) bekannt, berechneten Maße genutzt (siehe Hall-Beyer 2007).

Die vorgestellten Kategorien an Segmentierungsverfahren bieten zunächst einmal eine Möglichkeit, wie das Problem der Bildsegmentierung modelliert werden kann. Der eigentlichen Lösung des Problems kann diese Betrachtungsweise nur als Ausgangspunkt dienen. Auf Basis des Segmentierungsmodells ist es jedoch möglich mithilfe von Optimierungsansätzen zu einer Segmentation des Bildes zu kommen. So können histogramm-basierte Methoden als Auswahl der Schwellenwerte unter Minimierung der Segmentierungsfehler verstanden werden. Regionen-basierte Verfahren können beschrieben werden als Maximierung der Objektgröße unter der Nebenbedingung, dass die Heterogenität der Objekte einen gewissen Maximalwert nicht überschreiten darf. Kanten-basierten Verfahren entspricht in dieser Betrachtungsweise die Suche nach der Segmentation, in welcher möglichst viele relevante Kanten Bestandteil der Objektgrenzen sind. Bei modell-basierten Segmentierungsverfahren schließlich wird das Objektmodell so parametrisiert, dass die Anzahl der Fehlklassifikationen minimiert wird. Durch die Betrachtung der Segmentierung als Optimierungsproblem<sup>3</sup> ergibt sich eine Vielzahl an möglichen Verfahren, die zur Lösung eingesetzt werden können. Variiert man weiterhin Segmentierungsmodell, Heterogenitätskriterien und Anwendungsbereich, so wächst die Menge der möglichen Algorithmen weiter. Diese Tatsache spiegelt sich in der Zahl der Veröffentlichungen in diesem Forschungsbereich wider. So liefert die wissenschaftliche Suchmaschine Institute of Scientific Information (ISI) (2012) bei der Suche nach Artikeln, welche die Wörter „Image“ und „Segmentation“ im Titel tragen, fast 15000 Treffer. Trotzdem soll im Folgenden ein allgemeiner Überblick der existierenden Algorithmen zur Bildsegmentierung mit dem Verweis auf die weiterführenden Veröffentlichungen gegeben werden. Diese Auflistung ist nach dem zur Lösung verwendeten mathematischen Ansatz gruppiert. Sie erhebt keinen Anspruch auf Vollständigkeit.

- Verschiedene Aspekte der **Graphentheorie** werden oft zur Entwicklung von Segmentierungsverfahren genutzt. Der Grund dafür liegt unter anderem in der Tatsache begründet, dass ein Graph sehr intuitiv die Segmentation abbilden kann. Jeder Knoten des Graphs entspricht dabei einem Segment. Benachbarte Segmente sind mit einer Kante im Graphen verbunden, deren Gewicht ein Maß für die Ähnlichkeit der Segmente darstellt. Auf dieser Basis werden in Shi und Malik (2000) optimale Partitionen des Graphen mithilfe von minimal spannenden Bäumen erstellt, die der Algorithmus an jenen Stellen aufteilt, an denen die Unähnlichkeit zwischen benachbarten Knoten einen Schwellenwert übersteigt. Ebenfalls mit minimal spannenden Bäumen wird in Janakiraman u. a. (2007) ein Verfahren vorgestellt, dass ausgehend von dem Baum weitere Kanten hinzufügt, bis ein Kreis entsteht. Dieser Kreis wird dann zu einem Segment zusammengeführt. Ein drittes graphentheoretisches Segmentierungsverfahren auf Grundlage der minimal spannenden Bäume findet sich in Skurikhin (2008). Dieser Algorithmus erstellt den minimal spannenden Baum auf Grundlage einer Delaunay-Triangulierung und kombiniert dies mit Erkenntnissen aus der Gestalt-Theorie und Betrachtungen der Lagebeziehungen von Kanten (siehe Wertheimer 1923). Andere Veröffentlichungen nutzen die Ideen der

<sup>3</sup>Eine ausführliche Betrachtung der Bildsegmentierung aus Sicht der mathematischen Optimierung findet sich in Asano (2001)

Graphentheorie in Kombination mit Texturfiltern (siehe Galun u. a. 2003) oder kantenbasierten Heterogenitätskriterien (siehe Felzenszwalb und Huttenlocher 2004).

- Das Problem der Segmentierung kann auch als **Clustering** unter Nebenbedingungen verstanden werden. Ein direkter Weg der Segmentierung liegt daher in der Gruppierung der Pixel mit einem der vielen Clustering-Verfahren. Anschließend werden benachbarte Pixel zu demselben Segment gruppiert, wenn sie demselben Cluster angehören. Durch diese Analogie zwischen Segmentierung und Clustering steigt die Anzahl der verwendbaren Algorithmen noch einmal um ein Vielfaches. Für eine allgemeine Einführung zu Clustering-Algorithmen sei auf die sehr detaillierte Betrachtung in Duda u. a. (2001, S. 517–599) verwiesen. Für die Segmentierung wurden unter anderem das *Mean-Shift*-Verfahren eingesetzt (siehe Comaniciu und Meer 2002), das für jedes Pixel in einem räumlich-spektralen Eigenschaftsraum stationäre Punkte der Dichtefunktion sucht und diese als Segmentmittelpunkte zurückliefert. In Zoller und Buhmann (2007) wird ein Clustering-Algorithmus zur Segmentierung verwendet, der mithilfe des *Maximum-Likelihood*-Prinzips Gaußsche Mischverteilungen zur Kategorisierung verwendet. Dabei wird auch die Form der Objekte in die Analyse einbezogen. Ein Clustering-Ansatz, der die innere Heterogenität minimiert und gleichzeitig die Heterogenität zwischen benachbarten Segmenten maximiert wird in Bong und Rajeswari (2012) vorgestellt.
- Neben den im Clustering eingesetzten probabilistischen Methoden sind es vor allem Algorithmen auf Grundlage der **Markov-Theorie**, die für die Bildsegmentierung eingesetzt werden. Eine Markov-Kette ist dabei ein stochastischer Prozess, dessen Verteilung des nächsten Zustandes nur vom aktuellen Zustand abhängt und nicht von den zuvor angenommenen Zuständen. Eine solche Markov-Kette wird vom Algorithmus in Tu und Zhu (2002) genutzt, um Bilddaten zu segmentieren. Eine Erweiterung von Markov-Ketten bilden sogenannte verdeckte Markov-Ketten, bei denen der aktuelle Zustand nicht direkt beobachtbar, sondern über eine Verteilung einer Zufallsvariable bestimmt ist. Mithilfe eines solchen Modells wird in Tu u. a. (2005) eine hierarchisch Zerlegung des Bildes in seine Komponenten verfolgt das Verfahren aus Tu u. a. (2005). Die räumliche Variante der Markov-Ketten sind sogenannte Markov-Netzwerke, in denen die Verteilung einer Zelle stochastisch nur von der Verteilung in den benachbarten Zellen abhängt. Markov-Netzwerke werden unter anderem in Ueda u. a. (2000), Singaraju u. a. (2009), Feng u. a. (2010) und D’Elia u. a. (2003) zur Bildsegmentierung genutzt. Neben der Segmentierung finden Markov-Methoden in der Bildverarbeitung auch an anderen Stellen Verwendung. Eine generelle Einführung dazu wird in Li (2009) gegeben.
- In der Vergangenheit wurde kritisiert, dass die in der Bildverarbeitung verwendeten Methoden sehr oft mathematisch nicht ausreichend fundiert seien (siehe Aubert und Kornprobst 2006, S. vii-ix). Zumindest für den Bereich der Segmentierung existiert jedoch mit der **Theorie partieller Differentialgleichungen** ein Ansatz, der sehr wohl eine starke mathematische Grundlage besitzt. Eine wesentliche Arbeit zur Segmentierung mithilfe partieller Differentialgleichungen findet sich in Mumford und Shah (1989), worin das Problem der Segmentierung als Minimierung eines Energiefunktional formuliert wird. Dieser Ansatz führt laut Sonka u. a. (2008, S. 268) auf die Euler-Lagrange-Gleichung. Da sich diese Gleichung nicht in allen Fällen numerisch stabil verhält und stark von den Startwerten abhängt, wird in Donoser u. a. (2009) vor dem eigentlichen Algorithmus nach auffälligen Regionen gesucht, die anschließend als Eingabe der Energieminimierung genutzt werden. Die Segmentierungsmethoden mithilfe der Minimierung des Energiefunktional werden auch als *Active Contour Models* oder *Snakes* bezeichnet. Eine neuere Anwendung von *Snakes* für die Bildsegmentierung findet sich in Butenuth und Heipke (2012). Für eine weiterführende Betrachtung zur Bildsegmentierung mittels

partieller Differentialgleichungen sei auf Sonka u. a. (2008, S. 265–283) verwiesen.

- Neben vielen anderen Anwendungsbereichen werden **Methoden der künstlichen Intelligenz** auch für die Segmentierung eingesetzt. So werden in Perez und Olague (2007) genetische Algorithmen implementiert. Die Idee der genetischen Algorithmen ist ausgehend von einer Startsegmentierung durch zufällige Änderungen, sogenannte Mutationen, zu einer verbesserten Segmentierung zu gelangen. Dadurch, dass in einem Schritt eine Vielzahl an Varianten erzeugt wird, können über eine Bewertungsfunktion jeweils die besten Spezies einer Generation ausgewählt werden. Das entspricht dem evolutionären Prinzip des *survival of the fittest*. Darüber hinaus sind für die Bildsegmentierung auch neuronale Netzwerke im Einsatz, die versuchen, die Funktionsweise des menschlichen Gehirns nachzuahmen (siehe Marpu 2009; Xie u. a. 2011). Schließlich sei noch auf die Methoden der Fuzzytheorie verwiesen (siehe Brückmann 2004; Sonka u. a. 2008, S. 283–292). Mit diesen Algorithmen kann die Zugehörigkeit benachbarter Pixel zum selben Segment graduell modelliert werden. Die endgültige Segmentation enthält jedoch scharfe Segmentgrenzen, indem abschließend ein Schwellenwert gesetzt wird.
- Eine spezielle Art mathematischer Operationen ist der Typ der sogenannten **morphologischen Filter**. Im Gegensatz zu Faltungsoperatoren nutzen diese Filter eine besondere Algebra zur Transformation des Bildes. Mit diesen Filtern ist es möglich, ein Gradientenbild zu erzeugen, dass anschließend als Ausgangspunkt der sogenannten Wasserscheidentransformation dient. Meyer und Maragos (1999) gelten dabei als Begründer der morphologischen Wasserscheidentransformation. In Parvati u. a. (2009) wird ein Gradient über morphologische Filter gebildet. Anschließend wird die Segmentierung als Wasserscheidentransformation mit Markern als Startpunkten ausgeführt. O’Callaghan und Bull (2005) nutzen einen Texturfilter, der auf Wavelets beruht, und die anschließende Segmentierung mithilfe der Wasserscheidentransformation. Dabei wird eine Medianfilterung eingesetzt um zu vermeiden, dass Kanten im Bild zu starken Filterantworten führen. Abschließend wird mit graphentheoretischer Partitionierung das endgültige Segmentierungsergebnis erstellt.
- Durch die sinnvolle Zusammenfassung von benachbarten Pixeln zu Regionen ist es möglich, das Bild nicht mehr durch die Pixel und ihre Farbwerte, sondern durch die Segmente und ihre Eigenschaften zu repräsentieren. Diese Segmentrepräsentation des Bildes kann auch als **verlust-basierte Kompression** verstanden werden. Aus diesem Blickwinkel werden Methoden der Bildkompression auch für die Segmentierung eingesetzt. So nutzt der Algorithmus in Schwartzkopf u. a. (2002) eine entropie-basierte Kompressionsmethode zur Segmentierung von Mikroskopieaufnahmen. In Yang u. a. (2008) wird ein Kompressionsverfahren zur Segmentierung verwendet, das die Antwort eines Texturfilters quantisiert und anschließend mittels singulärer Gaußscher Mischverteilung versucht, die Pixel des Bildes in verschiedenen Gruppen zusammenzufassen. Diese Gruppen bilden dann die Segmente des Bildes.
- Alle bisher erwähnten Segmentierungsansätze benötigten abgesehen von der Wahl bestimmter Parameter keine Interaktion mit dem Nutzer. Es ist jedoch auch möglich, einen Benutzer aufzufordern vor der Ausführung der Segmentierung beispielhaft einzelne Objekte des Bildes zu umranden. Dieser Ansatz wird als **überwachte Segmentierung** bezeichnet. In Cesar u. a. (2005) wird ein Verfahren präsentiert, das die vollständige Segmentierung eines Bildes durch den Nutzer als Training für einen Modell-Graphen verwendet. Anschließend wird ein neues Bild per *Graph Matching* auf das Modell abgebildet und eventuell angepasst. Ein weiteres überwachtes Segmentierungsverfahren

findet sich in Noma u. a. (2012). Dieser Algorithmus nutzt dabei sehr simple Nutzereingaben, die sich auf das Markieren eines Objektes mit einzelnen Strichen beschränken. Darüber hinaus arbeitet dieses Verfahren iterativ, d.h. der Nutzer kann durch nachträgliches Hinzufügen von Trainingsgebieten die Segmentation verbessern. Überwachte Segmentierungsverfahren benötigen als Grundlage der Optimierungsprozedur ein Abstandsmaß, um die Ähnlichkeit der generierten Objekte mit den Trainingsgebieten zu bewerten. Diese Ähnlichkeitsmaße können darüber hinaus auch zur Evaluierung von Segmentationen eingesetzt werden (siehe Abschnitt 2.8.2).

Über das Modell und den eingesetzten Algorithmus hinaus wird das Ergebnis der Segmentierung auch dadurch bestimmt, ab wann ein Segment als homogen betrachtet wird. Dazu werden diverse Heterogenitätskriterien eingesetzt, die in folgende Kategorien gruppiert werden können:

- Farb-basierte Kriterien, z.B. der euklidische Abstand im Spektralraum oder die Varianz innerhalb eines Segments;
- Form-basierte Kriterien, welche beispielsweise die Glattheit der Außenkanten oder die Kompaktheit der Segmente abschätzen (siehe Baatz und Schäpe 2000);
- Kanten-basiert Kriterien, wie z.B. Sobel-, Canny- oder Laplacefilter (siehe Nischwitz und Haberäcker 2004);
- Texturkriterien, z.B. die GLCM-Maße aus Haralick u. a. (1973);
- Matchingkriterien, die z.B. aus der Faltung von Funktionen (siehe Sonka u. a. 2008, S. 238) oder der Hough-Transformation (siehe Sonka u. a. 2008, S. 212–221) abgeleitet werden.

In der Praxis wird in vielen Fällen nicht ein Kriterium allein verwendet, sondern es kommen mehrere Kriterien gleichzeitig zum Einsatz. Auch sind die Kriterien oftmals Bestandteil des Algorithmus.

Bisher wurden in diesem Abschnitt Segmentierungsverfahren aus einem allgemeinen Blickwinkel ohne Berücksichtigung des Anwendungsbereichs betrachtet. Zwar finden viele der besprochenen Verfahren auch im Bereich der Fernerkundung eine Verwendung. Jedoch wurde in Abschnitt 2.3.1 erläutert, dass die Bildsegmentierung für Fernerkundungsdaten einige Besonderheiten aufweist und die Verfahren daher oftmals einer Anpassung bedürfen. Aus diesem Grund soll im Folgenden ein Überblick der Segmentierungsverfahren für die Fernerkundung gegeben werden.

Einer der ersten Ansätze, der versucht, Fernerkundungsdaten objekt-basiert zu betrachten, findet sich in Kettig und Landgrebe (1976). Darin wird unter Verwendung von LANDSAT-Daten ein Algorithmus vorgestellt, der nach einer einzelnen Anwendung des *Split*-Verfahrens ein *Merge*-Verfahren implementiert. Das dabei eingesetzte Heterogenitätskriterium testet, ob zwei Segmente, die verbunden werden sollen, der gleichen Verteilung entstammen. Das Verfahren ist auch als *Extraction and Classification of Homogeneous Objects* (ECHO) bekannt und bildet noch heute einen Bestandteil des Softwarepakets *MultiSpec* (siehe Biehl und Landgrebe 2002). Inspiriert durch den ECHO-Ansatz wird in Cross u. a. (1988) ein *Split-Merge*-Verfahren für die Segmentierung von SPOT-Daten präsentiert, das neben den spektralen Eigenschaften der Pixel ein simples differenzen-basiertes Texturmaß verwendet. Eine weitere Pionierarbeit der Bildsegmentierung für die Fernerkundung findet sich in Bouman und Shapiro (1994). Darin wird ein Verfahren mit dem Titel *Sequential Maximum A Posteriori* (SMAP) präsentiert, das hierarchische Markov-Netzwerke einsetzt. In McCauley und Engel (1995) wird gezeigt, dass das SMAP-Verfahren bessere Ergebnisse als der ECHO-Ansatz liefert. Darüber hinaus

sind beide Verfahren einem pixel-basierten Ansatz überlegen.

Der nächste große Entwicklungsschritt im Bereich der objekt-basierten Bildverarbeitung für Anwendungen in der Fernerkundung vollzog sich mit der Veröffentlichung des *Multiresolution-Segmentation*-Algorithmus in Baatz und Schäpe (2000), der Eingang in die Software *eCognition* (siehe Definiens Imaging, Munich 2009) fand. Es ist damit eines der ersten Segmentierungsverfahren, das für die Analyse von Fernerkundungsdaten operationell genutzt werden konnte. Der *Multiresolution-Segmentation*-Algorithmus, der auf dem *Merge*-Ansatz beruht, findet aufgrund der sehr guten Qualität seiner Ergebnisse (siehe Marpu u. a. 2010) bis heute eine weite Verbreitung. Das Verfahren nutzt ein spektrales Heterogenitätskriterium auf Grundlage der Standardabweichung innerhalb eines Segments. Zusätzlich kann unter Verwendung eines Kriteriums für die Formhomogenität der Segmente festgelegt werden, ob die Außenkanten der Segmente möglichst glatt bzw. ob die Objektform möglichst kompakt sein sollte. Darüber hinaus existiert auch eine für die parallele Verarbeitung angepasste Variante des Algorithmus (siehe Happ u. a. 2010). Für weitere Details zur *Multiresolution Segmentation* sei auf Abschnitt 2.3.3 verwiesen.

Neben der Software *eCognition* werden objekt-basierte Ansätze auch in anderen kommerziellen Softwarepaketen zur Analyse von Fernerkundungsdaten implementiert. Zu nennen ist der objekt-basierte Ansatz in der Software *ENVI Fx* (siehe Solutions 2008), welcher ein nicht näher beschriebenes kanten-basiertes Segmentierungsverfahren verwendet. Für die Software *ENVI* existiert weiterhin eine kostenfreie Erweiterung namens *SCRM*, welche das *Watershed*-Verfahren mit vorheriger kantenerhaltender Glättung einsetzt (siehe Castilla u. a. 2008). Weiterhin wird für *ERDAS Imagine* (siehe ERDAS Imagine 2012) eine Erweiterung namens *Objective* vertrieben, die ein *Merge*-Verfahren nach Ruefenacht u. a. (2002) nutzt. Schließlich existiert mit der Erweiterung *APEX2000* für die Software *PCI Geomatica* (siehe PCI Geomatics 2000) laut Neubert u. a. (2008) ein in eine kommerzielle Fernerkundungssoftware implementiertes Verfahren, das nicht das gesamte Bild, sondern nur Objekte eines bestimmten Typs extrahiert. Die Firma *PCI* vertreibt seit 2008 allerdings auch die Software *eCognition*. Darüber hinaus sind Verfahren zur Objektextraktion in kostenlos erhältlichen Softwarepaketen verfügbar. Zum Beispiel existiert mit *SPRING* (siehe Instituto Nacional de Pesquisas Espaciais (INPE) 2012) ein umfangreiches System zur Segmentierung und objekt-basierten Klassifikation von Fernerkundungsdaten. Der darin enthaltene Segmentierungsalgorithmus arbeitet auf Basis eines *Merge*-Algorithmus mit euklidischem Abstandsmaß und abschließendem Entfernen kleiner Segmente (siehe Bins u. a. 1996). Die Software *EDISON* (siehe Christoudias u. a. 2002) bietet einen Segmentierungsalgorithmus auf Grundlage des *Mean-Shift*-Verfahrens mit Kantendetektion nach Comaniciu und Meer (2002). Das Programm wurde jedoch nicht in erster Linie für die Analyse von Fernerkundungsdaten entwickelt. Ein ebenfalls nicht speziell für die Fernerkundung entwickeltes jedoch kostenfreies Bildanalysesystem ist das Programm *ImageJ* (siehe Rasband 1997). Für diese Software existieren eine Reihe von Erweiterungen zur Bildsegmentierung (National Institutes of Health (NIH) 2003) auf Basis verschiedener Algorithmen.

Auf Grundlage der kostenpflichtigen Bildverarbeitungssoftware *HALCON* (siehe MVTec 2011), die u. a. für industrielle Anwendungen entwickelt wurde, arbeitet das Segmentierungsverfahren *HalconSEG*. Die Ideen zu diesem Verfahren beruhen auf einem *Merge*-Algorithmus mit Kantenoperator, der in Herold (2005) präsentiert wurde.

Abschließend sei erwähnt, dass es für die Segmentierung von Daten der Radarfernerkundung spezielle Softwarepakete existieren. Zu nennen sind *InfoPACK* (siehe N. A. Software Limited 2010) und *SegSAR* (siehe Arouja Sousa 2009). Der Rahmen dieser Arbeit beschränkt sich jedoch auf die Betrachtung von Verfahren für optische Fernerkundungsdaten.

Außer den in operationell nutzbaren Systemen eingesetzten Segmentierungsalgorithmen wurden in der wissenschaftlichen Literatur eine Reihe weiterer Verfahren für die Segmentierung von Fernerkundungsdaten vorgeschlagen, implementiert und teilweise auch evaluiert. Im Fol-

genden werden einige Segmentierungsverfahren für Fernerkundungsdaten genannt. Die Kategorisierung findet dabei gemäß der zugrunde liegenden Herangehensweise statt. Eine allgemeine Übersicht zu diesem Thema findet sich auch in Dey u. a. (2010).

Fernerkundungsaufnahmen sind nicht dadurch charakterisiert, dass einzelne Objekte sich vor einem homogenen Hintergrund abzeichnen, sondern dass sie flächendeckend aus verschiedenen Objekten bestehen. Aus diesem Grund sind regionen-basierte Methoden eine direkte Art der Lösung des Segmentierungsproblems und werden in vielen Varianten in der wissenschaftlichen Literatur vorgeschlagen. Der simpelste regionen-basierte Segmentierungsalgorithmus, das *Merge*-Verfahren, wird sehr häufig angewendet. In Bins u. a. (1996) und Gofman (2006) findet es in Kombination mit dem euklidischen Abstand Anwendung. Auch Tilton (2003) betrachtet eine Reihe verschiedener Abstandsmaße zur Segmentierung mittels *Merge*-Algorithmus. In Benz u. a. (2004) wurde der bereits erwähnte *Multiresolution-Segmentation*-Algorithmus genutzt, ein *Merge*-Verfahren unter Zuhilfenahme eines Heterogenitätskriterium, das sich aus einer Mischung von Farb- und Formeigenschaften ergibt. Als Abwandlung des *Merge*-Verfahrens findet auch das *Split-Merge*-Verfahren Anwendung im Bereich der Fernerkundung. Beispielsweise nutzt Lucieer (2004) diesen Ansatz zusammen mit einem multivariaten Texturfilter.

Ein ebenfalls regionenorientiertes Verfahren ist der *Mean-Shift*-Algorithmus, eigentlich ein Algorithmus zum Gruppieren von Objekten im Merkmalsraum. Canty (2009, S. 303–304) nutzt dieses Verfahren zur Segmentierung, indem der Farbraum um eine Dimension erweitert wird und dadurch die räumliche Nähe im Bild auf die Nähe im Merkmalsraum abbildet. In Chen u. a. (2010) werden Luftbilder mithilfe des *Mean-Shift*-Algorithmus auf Grundlage von Wavelet-Komponenten segmentiert. González u. a. (2009) nutzt den *Mean-Shift*-Algorithmus zur Segmentierung von Wolken in meteorologischen Satellitenbildern.

Auch graphentheoretische Algorithmen finden Eingang in die regionen-basierte Segmentierung von Fernerkundungsdaten. So verwendet z.B. Marpu (2009) ein zweistufiges Verfahren angelehnt an das Verfahren der minimal spannenden Bäume aus Shi und Malik (2000). In einem ersten Schritt wird bei diesem Algorithmus mithilfe des *Merge*-Verfahrens eine initiale Segmentierung mit Segmenten kleiner Fläche erzeugt. Im zweiten Schritt wird ein minimal spannender Baum erzeugt, der an „schwachen“ Kanten aufgeteilt wird und dadurch in der endgültigen Segmentierung resultiert.

Neben den regionen-basierten Verfahren können auch Segmentierungsalgorithmen eingesetzt werden, welche die Umrisse der Objekte detektieren und daraus eine Segmentation des Bildes ableiten. Ein Beispiel für ein solches kanten-basiertes Verfahren ist die in Grazzini u. a. (2010) vorgestellte Herangehensweise. Dieser Algorithmus erzeugt mittels eines multispektralen Kantenfilters und des Delaunay-Verfahrens eine ungleichförmige Triangulierung des Bildes. Anschließend werden die Polygone mit Prinzipien der Gestalttheorie zu Segmenten zusammengefasst.

Das wohl am häufigsten in der Fernerkundung eingesetzte kanten-basierte Segmentierungsverfahren ist die Wasserscheidentransformation. In Li und Xiao (2007) wird die Wasserscheidentransformation auf einem Gradientenbild ausgeführt, das durch eine multispektrale morphologische Filterung generiert wird. Da der Algorithmus zur Übersegmentierung neigt, werden in der Literatur verschiedene Verfahren zur weiteren Verarbeitung vorgeschlagen. Lefebvre u. a. (2008) setzt eine Wasserscheidentransformation in einem initialen Schritt ein, um ein übersegmentiertes Zwischenergebnis zu erhalten. Darauf aufbauend wird das Bild mittels Wavelets transformiert. Anschließend wird auf Grundlage der Waveletkoeffizienten ein *Merge*-Verfahren gestartet, das die endgültige Segmentation liefert. Generell sind kanten-basierte Verfahren für Fernerkundungsdaten aber nur bedingt geeignet, weil diese Algorithmen für Bildrauschen besonders anfällig sind. Des weiteren wird die Wasserscheidentransformation Anwendung zur Segmentierung von Fernerkundungsdaten in Pesaresi und Benediktsson (2001) und Aytekin und Ulusoy (2011) eingesetzt.

Auch wenn in Cushnie (1987) vorgeschlagen wird, durch Filterung texturierte Bereiche in Fer-



nerkundungsdaten zu entfernen, kann Texturinformation trotzdem zur Segmentierung verwendet werden. So nutzt z.B. Kristinsdóttir (2008) eine Filterung mithilfe invarianter Momentmethoden. Die Filterantwort wird dabei anschließend mit dem *Multiresolution-Segmentation*-Algorithmus segmentiert. In Ryherd und Woodcock (1996) kommt eine spezielle Varianzfilterung des Bildes zum Einsatz. Anschließend wird ein *Merge*-Algorithmus eingesetzt. Schließlich lassen sich auch Wavelet-basierte Texturfilter als Grundlage der *Multiresolution Segmentation* verwenden, wie in Van Coillie u. a. (2008a) gezeigt wird.

Im Anwendungsbereich der Fernerkundung finden neben jenen Verfahren, die eine Segmentation gemäß der Gleichungen 2.1 erzeugen, auch spezielle Algorithmen Anwendung. In Li u. a. (2011) etwa wird ein Segmentierungsverfahren vorgestellt, das zur Detektion von Flugzeugen keine flächendeckende Segmentation erzeugt, sondern nur einzelne Bereiche hervorhebt. Lizarazo und Elsner (2009) schlägt eine unscharfe Segmentierung mit Methoden der Fuzzytheorie vor. Dabei entstehen Objekte ohne scharfe Objektgrenzen. Da der in Abschnitt 2.1 beschriebene Ablauf relativ strenge Anforderungen an die Art der Segmentierung stellt, können diese Verfahren im Folgenden nicht weiter berücksichtigt werden.

Neben vollautomatischen Segmentierungsansätzen werden in der Literatur auch Verfahren vorgestellt, die eine Nutzerinteraktion umfassen. Esch u. a. (2008) verwendet gemäß dieses Ansatzes den *Multiresolution-Segmentation*-Algorithmus und eine regel-basierte Klassifikation, um die Segmentierungsparameter zu bestimmen. Die resultierende Segmentation weist anschließend einen geringeren Fehler auf als bei einem Verfahren ohne Einwirken des Nutzers. Andere Verfahren wie z.B. jene in O'Neil-Dunne u. a. (2009) und Hofmann u. a. (2008) verwenden mehrere Iterationen der Prozessschritte Segmentierung, Klassifikation und Aufteilung der Segmente. Dadurch wird eine Segmentierung erreicht, die klassenspezifisch bessere Ergebnisse liefert. Jedoch wohnt diesem Ansatz das Problem inne, dass die Regelsätze zur Klassifizierung speziell auf einzelne Bilddaten zugeschnitten sind und kaum auf andere Daten übertragen werden können. Außerdem birgt die starke Nutzerinteraktion das Risiko, dass die Ergebnisse nicht stabil reproduziert werden können, weil das Ergebnis stark von subjektiven Entscheidungen des Nutzers abhängt.

Auch für die Radarfernerkundung existieren in der Literatur vielfältige Verfahren zur Segmentierung. Da dieses Gebiet jedoch nicht der Fokus der vorliegenden Arbeit ist sei beispielhaft auf die Arbeiten von Wu und Leahy (1993), Xiaochun und Jing (2004) und Ferro-Famil u. a. (2003) verwiesen.

Die große Anzahl der angesprochenen Verfahren wirft die Frage auf, welches Verfahren für die objekt-basierte Änderungsdetektion am besten geeignet ist. Ein großer Kritikpunkt am momentanen Forschungsstand ist die Tatsache, dass die Funktionsweise vieler Verfahren zwar im Detail erläutert wird, jedoch keine ausreichende Bewertung der Qualität ihrer Ergebnisse stattfindet. Zum einen liegt dies daran, dass in zur Erläuterung eines Verfahrens oft nur ein bestimmter für die jeweilige Quelle spezifischer Datensatz als Beispiel herangezogen wird. Im Gegensatz zu Standardalgorithmen der Bildverarbeitung, die häufig das sogenannten Lena-Bild (siehe Rosenberg 1996) als Beispiel verwenden, existiert in der Fernerkundung kein klassisches Referenzbeispiel. Andererseits gibt es auch keine einheitliche Methodik für die Evaluierung von Segmentierungsverfahren. Nichtsdestoweniger existieren einige vergleichende Studien zur Evaluierung der Segmentierungsgüte in Meinel und Neubert (2004), Neubert u. a. (2008) und Marpu u. a. (2010). Eine weiterführende Diskussion dieses Themas findet sich in Abschnitt 2.8.2. In allen drei Studien wurde betont, dass der *Multiresolution-Segmentation*-Algorithmus gute Ergebnisse liefert. Darüber hinaus zeigt (siehe Niemeyer u. a. 2008), dass der Algorithmus mit Erfolg im Bereich der objekt-basierten Änderungsdetektion eingesetzt werden kann. Aus diesen Gründen wird in der vorliegenden Arbeit das Verfahren der *Multiresolution Segmentation* eingesetzt. In den folgenden Abschnitten wird gezeigt, welche Voraussetzungen für den Einsatz des Algorithmus im Bereich der objekt-basierten Änderungsdetektion notwendig sind.

### 2.3.3. Multiresolution Segmentation

#### Ablauf des Verfahrens

Der *Multiresolution-Segmentation-Algorithmus* (siehe Baatz und Schäpe 2000; Baatz u. a. 2003) ist ein regionen-basiertes Segmentierungsverfahren, das auf dem *Bottom-Up*-Prinzip beruht. Iterativ werden benachbarte Objekte miteinander verbunden, wenn sie einander ähnlich sind. Findet der Algorithmus keine ähnlichen Objektpaare mehr, ist die endgültige Segmentation erreicht. Die Ähnlichkeit wird dabei mithilfe eines Heterogenitätsmaßes quantifiziert, das neben der spektralen Unterschiedlichkeit auch die entstehende Form der Objekte berücksichtigt.

Die Einfachheit des Algorithmus ist neben der Segmentierungsgüte ein wichtiges Argument für die Verwendung im Rahmen dieser Arbeit. Um jedoch den Algorithmus im Bereich der Änderungsdetektion einzusetzen, sind Anpassungen nötig. Diese Anpassungen bedürfen eines tiefgreifenden Verständnisses des Originalalgorithmus. Daher widmet sich dieser Unterabschnitt der Erläuterung des Verfahrens der *Multiresolution Segmentation* für Einzelbilder.

Im Detail beginnt das Verfahren damit, jedes Pixel des Bildes als ein Objekt zu betrachten. Dazu werden für jedes Objekt die Beziehungen zu den Nachbarobjekten etabliert damit das Verfahren mögliche Segmentpaare auswählt, die verbunden werden können. Im Rahmen dieser Arbeit wird eine sogenannte 4-Nachbarschaft verwendet, d.h. jedes Pixel, das sich nicht am Rand des Bildes befindet, besitzt einen nördlichen, einen südlichen, einen östlichen und einen westlichen Nachbarn.

Anschließend wählt das Verfahren ein zufälliges Segment  $s_1$  als Startpunkt aus<sup>4</sup> und versucht es mit dem ähnlichsten Nachbarn  $s_2$  zusammenzufügen. Ist in der Nachbarschaft von  $s_2$  auch  $s_1$  das Segment, welches am ähnlichsten ist, so werden  $s_1$  und  $s_2$  miteinander verbunden. Andernfalls wählt der Algorithmus  $s_2$  als Startsegment und beginnt die Suche erneut. Diese Suchstrategie nennt man *Local Mutual Best Fitting*. Sie führt dazu, dass jenes Segmentpaar verbunden wird, das lokal die größte Ähnlichkeit besitzt. Die Strategie ist analog zum Finden eines lokalen Optimums zu sehen und nimmt Anleihen beim Verfahren des steilsten Abstieges (siehe Wikipedia 2012). In Abbildung 2.3 ist diese Art der Suche schematisch dargestellt.

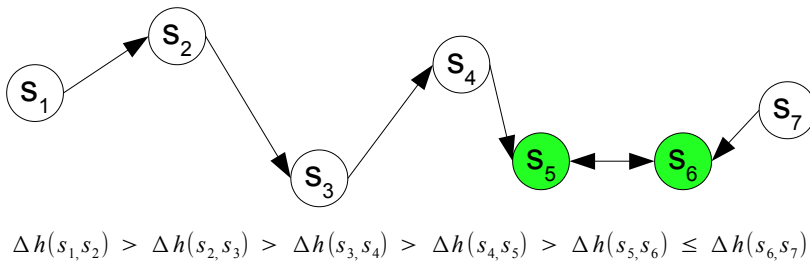


Abb. 2.3.: Strategie des *Local Mutual Best Fitting*.

Andere Suchstrategien sind das *Local Best Fitting* und das *Global Best Fitting*. Beim *Local Best Fitting* wird das Startsegment direkt mit dem ähnlichsten Nachbarn verbunden, wenn das Maß für die Ähnlichkeit unterhalb eines Schwellenwertes liegt. Dieser Ansatz hat den

<sup>4</sup>In Baatz und Schäpe (2000) wird zur Auswahl der Startsegmente die Nutzung von sogenannten Dithermatrizen nach Abmayr (1994) vorgeschlagen. Im Rahmen dieser Arbeit konnte jedoch nicht ermittelt werden, wie dieses Verfahren zu realisieren ist. Darüber hinaus wird auch in Baatz u. a. (2003) die Auswahl der Startsegmente über ein stochastisches Verfahren als Möglichkeit erwähnt. Aus den genannten Gründen wird für die Implementierung des *Multiresolution-Segmentation-Algorithmus* in dieser Arbeit eine stochastische Auswahl der Startsegmente implementiert.

Vorteil, dass durch die Wahl des Startsegments der Ort des Segmentwachstums bereits festgelegt ist und nicht durch eine wenn auch lokal geartete Suche verändert wird. Dabei kann es jedoch vorkommen, dass durch die zufällige Auswahl des Startsegment in einem sehr heterogenen Bildbereich liegt und daher das Ziel, möglichst homogene Bildbereiche zu segmentieren, nicht erreicht wird. Das Prinzip des *Global Best Fitting* verbindet jenes Segmentpaare, das unter allen Segmentpaaren im Bild die größte Ähnlichkeit aufweist. Dadurch wird garantiert, dass heterogene Bereiche des Bildes nicht gegenüber homogenen Bereichen bevorzugt werden. Jedoch kann ein gleichmäßiges Segmentwachstum bei diesem Vorgehen nicht garantiert werden. Aus den genannten Gründen wird das Prinzip des *Local Mutual Best Fitting* für den *Multiresolution-Segmentation*-Algorithmus verwendet. Für die Implementierung des Algorithmus wurde jedoch ein zusätzlicher Parameter Mutual Tolerance eingeführt, der stetig zwischen 0 und 1 variiert werden kann und der festlegt, wie strikt das Paradigma des *Local Mutual Best Fitting* eingehalten werden soll. Zur Verdeutlichung der Wirkungsweise des Parameters sei die Situation gegeben, dass ausgehend von einem Segment  $s_1$  das Segment  $s_2$  der bestpassende Nachbar ist. Ein Wert  $p$  des Parameters führt in dieser Situation dazu, dass  $s_2$  auch dann mit  $s_1$  verbunden wird, wenn es in der Nachbarschaft von  $s_2$  einen gewissen Anteil von  $p$  gibt, der besser zu  $s_2$  passen würde, als es das Segment  $s_1$  tut. Für Werte nahe 0 kommt es daher zu einer strengen Interpretation des *Local Mutual Best Fitting*. Bei Werten nahe 1 spielt das Kriterium der gegenseitig bestpassenden Segmente kaum eine Rolle mehr und es dominiert eher das Prinzip des *Local Best Fitting*.

Die Auswahl des zu fusionierenden Segmentpaares beruht auf einem Maß für die Unterschiedlichkeit der Segmente. Dazu kommt ein sogenanntes Heterogenitätsmaß  $h$  zum Einsatz. Zwei Segmente  $s_1$  bestehend aus  $n_1$  Pixeln und  $s_2$  bestehend aus  $n_2$  Pixeln werden dabei als umso unterschiedlicher betrachtet, je größer die gewichtete Differenz  $\Delta h$  zwischen der Heterogenität des neu entstehenden Segments und den Ausgangsheterogenitäten ist. Die Suche nach dem optimalen Segmentpaar lässt sich damit formalisieren als

$$\Delta h(s_1, s_2) = (n_1 + n_2)h(s_1 \cup s_2) - (n_1 h(s_1) + n_2 h(s_2)) \rightarrow \min. \quad (2.4)$$

Zusätzlich wird zur Erfüllung des Homogenitätskriteriums  $H$  gefordert, dass die Heterogenität des neu entstehenden Segments unterhalb eines Schwellenwertes  $T$  liegt, d.h. es muss gelten

$$H(s_1 \cup s_2) = \text{wahr} \iff \Delta h(s_1, s_2) < T. \quad (2.5)$$

Für ein Segment  $s$  bestehend aus  $n$  Pixeln kann  $h(s)$  als gewichtete Summe aus einem Farbanteil  $h_c$  und einem Formanteil  $h_s$  berechnet werden, d.h.

$$h(s) = w_c h_c(s) + w_s h_s(s). \quad (2.6)$$

Die Gewichte für Farbe  $w_c$  und Form  $w_s$  sind Parameter des Verfahrens und werden so festgelegt, dass für ihre Summe gilt

$$w_c + w_s = 1 \text{ mit } w_c, w_s \geq 0. \quad (2.7)$$

Die Farbheterogenität  $h_c$  eines Segments  $s$  bestehend aus den Pixeln  $\mathbf{x}_1, \dots, \mathbf{x}_n$  wird als gewichtete mittlere empirische Standardabweichung der Pixel innerhalb der  $k$  Bildkanäle definiert, d.h.

$$h_c(s) = \sum_{j=1}^k w_j \hat{\sigma}_j, \quad (2.8)$$

mit für den geschätzten Mittelwert des Segments im  $j$ -ten Kanal

$$\hat{\mu}_j = \frac{1}{n} \sum_{i=1}^n x_{i,j}, \quad (2.9)$$

und

$$\hat{\sigma}_j^2 = \frac{1}{n} \sum_{i=1}^n (x_{i,j} - \bar{\mu}_j)^2, \quad (2.10)$$

für die unkorrigierte, geschätzte Varianz des Segments im  $j$ -ten Kanal.  
Die nicht negativen Gewichte werden mittels

$$\sum_{j=1}^k w_j = 1 \quad (2.11)$$

normiert und können ansonsten frei gewählt werden. In den Standardeinstellungen des Algorithmus werden alle Kanäle gleich gewichtet.  
Die Formheterogenität von  $s$  ergibt sich wiederum durch die gewichtete Summe der Glattheit  $h_{smooth}$  und Kompaktheit  $h_{comp}$  als

$$h_s(s) = w_{smooth} h_{smooth}(s) + w_{comp} h_{comp}(s), \quad (2.12)$$

wobei die Gewichte erneut normiert werden durch

$$w_{smooth} + w_{comp} = 1 \text{ mit } w_{smooth}, w_{comp} \geq 0. \quad (2.13)$$

Die Kompaktheit eines Segments ist definiert als

$$h_{comp}(s) = \frac{l}{\sqrt{n}}. \quad (2.14)$$

Motiviert wird die Nutzung der Kompaktheit dadurch, dass im stetigen Falle die rechte Seite der Gleichung 2.14 gemäß der isoperimetrischen Ungleichung (siehe Osserman 1978) minimal für ein Segment in Form eines Kreises.

Für die Glattheit gilt

$$h_{smooth}(s) = \frac{l}{b}. \quad (2.15)$$

Dabei ist  $l$  der Umfang des Segments  $s$  und  $b$  der Umfang des kleinsten Rechtecks, das  $s$  beinhaltet. Das Verhältnis der Umfänge von Segment und kleinstem umfassenden Rechteck ist für Segmente mit glatten Begrenzungslinien kleiner als für jene mit sehr rauen. Durch die Verwendung der Glattheit und Kompaktheit als Maße für die Homogenität eines Segments wird erreicht, dass der Algorithmus Objekte extrahiert, die dem Ergebnis einer manuellen Segmentierung ähneln. Dies liegt in der Tatsache begründet, dass Menschen dazu neigen, Gesehenes zu abstrahieren und somit die Außenkanten eines Objektes nicht nur versuchen so exakt wie möglich abzubilden, sondern auch eine Repräsentation wählen, welche die wesentlichen Eigenschaften des Objekts hervorhebt (siehe Wertheimer 1923).

Mit dem beschriebenen Suchverfahren und dem Maß für die Heterogenität  $h$  verbindet der Algorithmus solange Segmentpaare, bis kein Segmentpaar gefunden werden kann, dessen Heterogenität unterhalb des Schwellenwertes  $T$  liegt.

Aufgrund der Beschreibung des Algorithmus wird klar, dass bereits kleine Änderungen an den Daten wesentliche Modifikationen des Resultats nach sich ziehen können. Diese mangelnde Stabilität des Algorithmus in Kombination mit einer Referenzimplementierung, deren Quellcode nicht öffentlich zugänglich ist, bilden die zwei wesentlichen Nachteile des *Multiresolution-Segmentation*-Algorithmus für eine Verwendung im Rahmen des in Abschnitt 2.1 beschriebenen Verfahrens zur objekt-basierten Änderungsdetektion.

Andererseits existiert bis heute kein universell einsetzbares stabiles Segmentierungsverfahren für multispektrale Fernerkundungsdaten, und die Beschreibung des *Multiresolution-Segmentation*-Algorithmus in Baatz und Schäpe (2000) zeigt ein relativ simples Verfahren, das für die

objekt-basierte Änderungsdetektion angepasst werden kann. In Bezug auf die Stabilität sei angemerkt, dass trotz zufälliger Auswahl der Ausgangssegmente zumindest eine Reproduzierbarkeit bei gleichen Eingabedaten gegeben ist. Der Algorithmus beginnt nämlich mit der Initialisierung des Pseudozufallszahlengenerators. Dadurch wird immer dieselbe Folge von Zufallszahlen generiert und somit werden bei gleichen Eingabedaten auch dieselben Startpunkte gewählt.

Ein weiterer Aspekt der für die Verwendung des Algorithmus spricht ist die Güte der Segmentierung. In verschiedenen Studien zur Evaluierung der Segmentierungsergebnisse schneidet der *Multiresolution-Segmentation*-Algorithmus überdurchschnittlich gut ab (siehe Marpu u. a. 2010; Neubert u. a. 2008).

Schließlich sei erwähnt, dass der *Multiresolution-Segmentation*-Algorithmus in der Lage ist, verschiedene Arten von Eingabedaten zu kombinieren (*multi source*), unterschiedliche Objekttypen gleichzeitig, d.h. innerhalb eines Bildes, extrahieren kann (*multi method*) sowie in mehreren Generalisierungsebenen Objekte segmentiert. Daher erfüllt er die in Abschnitt 2.3.1 aufgestellten Anforderungen an einen Segmentierungsalgorithmus.

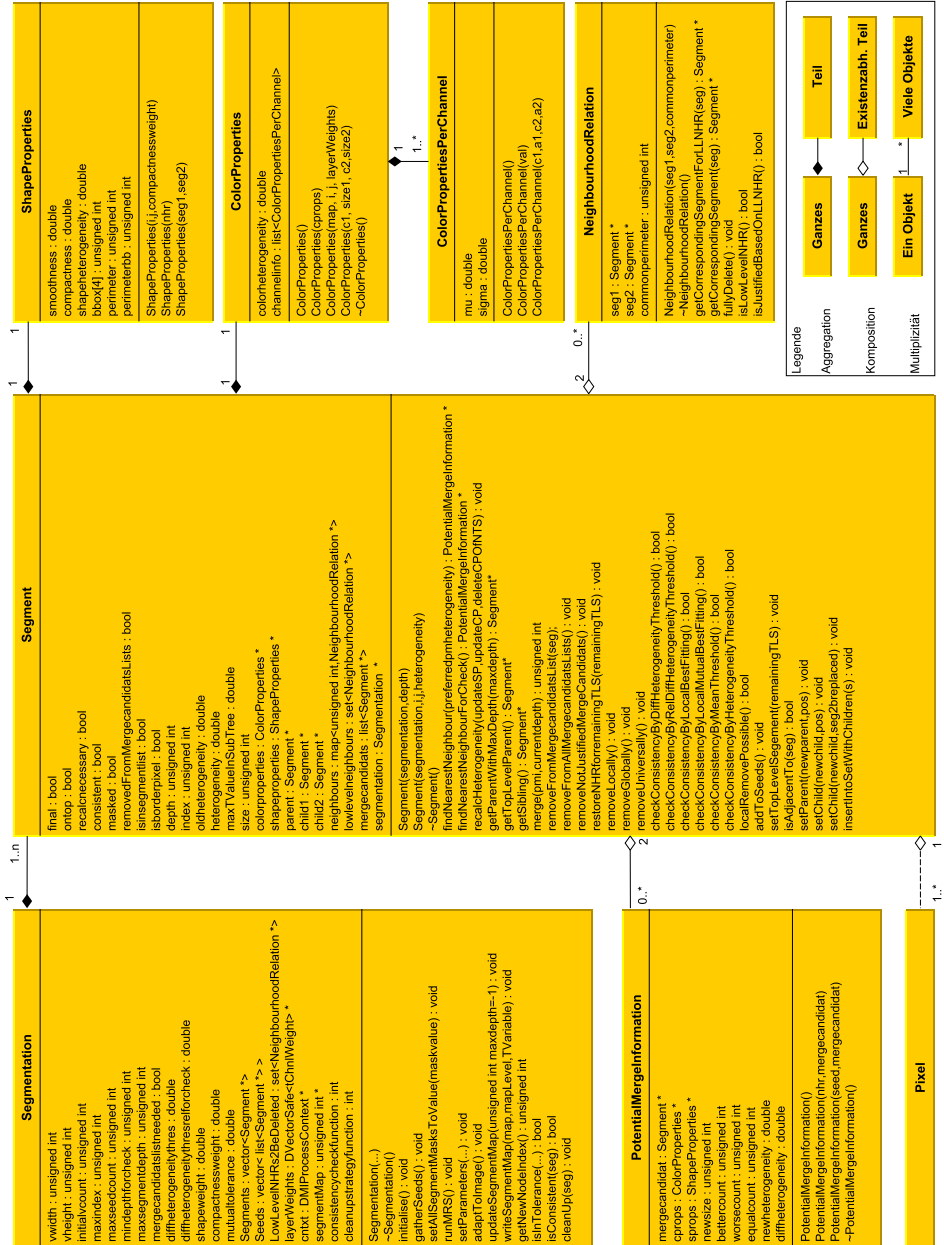
Auch wenn der *Multiresolution-Segmentation*-Algorithmus einige Aspekte beinhaltet, die für eine Verwendung im Rahmen der objekt-basierten Änderungsdetektion nachteilig sind, überwiegen doch die Vorteile des Verfahrens. Aus diesem Grund wird der Algorithmus innerhalb dieser Arbeit auf eine Anwendung für die objekt-basierte Änderungsdetektion erweitert und angepasst.

## Reimplementierung des Algorithmus

Das Verfahren der *Multiresolution Segmentation* wurde im Rahmen der Software *eCognition* hinsichtlich Speicherbedarf und Rechenzeit in einer sehr effizienten Art implementiert. Um jedoch den Algorithmus auf das Problem der Änderungsdetektion anzupassen, ist eine Reimplementierung nötig. Ein Rückgriff auf die in der Software *eCognition* enthaltene Version ist nicht möglich, da zwar eine Programmierschnittstelle zu *eCognition*, ein sogenanntes *Software Development Kit* (SDK), existiert, aber das MRS-Verfahren darüber nicht angesteuert werden kann.

Die Reimplementierung des *Multiresolution-Segmentation*-Algorithmus wurde im Rahmen dieser Arbeit in der Programmiersprache C++ als Plugin für *eCognition* erstellt. Dazu kommt eine objektorientierte Architektur zum Einsatz, welche in Abbildung 2.4 dargestellt ist. Das Modell geht von einer Repräsentation der Segmentation in der gleichnamigen Klasse aus. Diese Klasse enthält eine Sammlung von Objekten des Typs **Segment** sowie Attribute und Methoden zur Steuerung des Segmentierungsprozesses. Die Klasse **Segment** wiederum enthält einerseits elementare Eigenschaften wie etwa die Attribute **final** und **ontop**, die das jeweilige Segment hinsichtlich seines Status charakterisieren. Andererseits werden in jedem Segmentobjekt weitere Unterobjekte der Typen **ColorProperties** bzw. **ShapeProperties** referenziert, welche die Farb- respektive Formeigenschaften speichern. Die Farbeigenschaften spiegeln dabei die Farbheterogenität aus Gleichung 2.8 wider, indem die empirischen Werte der Standardabweichung  $\hat{\sigma}_j$  (siehe Gleichung 2.10) und des Mittelwerts  $\hat{\mu}_j$  (siehe Gleichung 2.9) für jeden Kanal in einem Unterobjekt vom Typ **ColorPropertiesPerChannel** mit den Attributen **mu** und **sigma** abgelegt werden.

Objekte vom Typ **Segment** repräsentieren entweder einzelne Pixel oder entstehen durch die Fusion zweier existierender Segmente. Dadurch kann ein einzelnes Segment auch als binärer Baum verstanden werden. Diese Überlegung wird im nächsten Abschnitt eine wichtige Rolle spielen. Um diese Baumrepräsentation zu realisieren, sind jedem Segmentobjekt zwei Eigenschaften zugeordnet: eine Referenz auf ein Elternsegment im Attribut **parent** sowie zwei Referenzen auf die Kindsegmente in den Attributen **child1** und **child2**. Im Fall, dass die

Abb. 2.4.: Klassendiagramm zur Implementierung des *Multiresolution-Segmentation*-Algorithmus.

Referenzen auf die Kindsegmente nicht definiert sind, handelt es sich um ein Pixelsegment. Ist die Referenz auf das Elternsegment undefiniert, liegt ein Segment der obersten Hierarchieebene vor.

Durch diese Datenstruktur wird erstens festgehalten, aus welchen Segmenten ein Segment entstanden ist. Zweitens kann durch diese Art der Informationshaltung über die Pixelsegmente im Segmentbaum festgestellt werden, welche Pixel zu einem Segment gehören. Dies ist für die Erzeugung einer Segmentkarte wichtig. Die Liste der zu einem Segment gehörenden Pixel wird aus den genannten Gründen nur implizit gespeichert, was im Klassendiagramm in Abbildung 2.4 durch eine gestrichelte Linie dargestellt ist. Für Pixelsegmente ergibt sich das zugehörige Pixel über den Segmentindex.

Die Nachbarschaftsbeziehungen zwischen Segmentobjekten werden über die Klasse **NeighbourhoodRelation** modelliert. In einem Objekt vom Typ **NeighbourhoodRelation** wird neben jeweils zwei Referenzen auf die betreffenden Segmentobjekte in den Attributen **seg1** und **seg2** auch die Länge der gemeinsamen Grenze der Segmente in der Eigenschaft **commonperimeter** gespeichert. Diese Klasse spielt bei der Implementierung des Algorithmus und insbesondere während des Verbindens zweier Segmente eine wesentliche Rolle. In jedem Objekt vom Typ **Segment** wird nämlich eine Liste mit Objekten vom Typ **NeighbourhoodRelation** gespeichert.

Der eigentliche funktionale Teil des *Multiresolution-Segmentation*-Algorithmus besteht aus zwei wesentlichen Bestandteilen, die im Folgenden erläutert werden sollen: der Suche nach einem zur Verbindung geeigneten Segmentpaar und dem Ablauf der Fusion selbst.

Die Implementierung der Suche nach zur Verbindung geeigneten Segmentpaaren basiert wesentlich auf einer sortierten Liste aller im Bild vorhandenen Segmente der obersten Hierarchieebene, die vom Algorithmus nicht als final eingestuft wurden. Das Attribut **Seeds** der Klasse **Segmentation** speichert diese Ausgangssegmente. Anfänglich werden in diese Liste alle Pixelsegmente eingefügt und mit der Funktion **random\_shuffle** in eine zufällige Reihenfolge gebracht. Dabei kann die Reproduzierbarkeit der Segmentation durch die Wahl der Parameter des Pseudozufallszahlengenerators erreicht werden. Im weiteren Verlauf des Algorithmus wird jedoch bei jedem Verbindungsschritt das neu entstehende Segment in diese Liste eingefügt. Die beiden Ausgangssegmente der Verbindung werden hingegen aus der Liste entfernt. Dadurch speichert diese Liste die Ausgangssegmente des Wachstumsprozesses.

Als Sortierkriterium der Liste nutzt der Algorithmus die sogenannte Tiefe, d.h. die maximale Anzahl an Verbindungsschritten, die jedes Pixelsegment durchlaufen musste, um Teil des jeweiligen Segments der obersten Hierarchieebene zu werden. Gespeichert wird die Tiefe im Attribut **depth** der Klasse **Segment**. Mithilfe dieser Sortierung der Liste kann ein gleichmäßiges Wachstum der Segmente erreicht werden, da zu jedem Zeitpunkt versucht wird, jene Bereiche des Bildes zu betrachten, die bisher möglichst selten Bestandteil eines Verbindungsschrittes waren. Durch diesen Ansatz wird ein Defizit des *Local Mutual Best Fitting* behoben. Interessant im Zusammenhang mit der Diskussion über die Gleichmäßigkeit des Wachstums ist die Tatsache, dass es für die Anzahl der Segmentfusionen (und daraus folgend für die Laufzeit des Algorithmus) nicht entscheidend ist, in welcher Reihenfolge die  $n$  Pixel eines Segments aggregiert werden. Denn bei  $n$  Ausgangssegmenten und einem daraus entstehenden Endsegment gibt es genau  $n - 1$  Verbindungsschritte, da bei jeder Fusion die Anzahl der Segmente in der obersten Hierarchieebene um eins reduziert wird unabhängig davon, welche der  $n$  Segmente miteinander in welcher Reihenfolge verbunden werden. Auf die Tiefe des Segments hat der Ablauf des Prozesses jedoch sehr wohl Einfluss. Denn bei sehr ungleichmäßigem Wachstum wird in jedem Verbindungsschritt ein einzelnes Pixel mit dem größten Segment verbunden, was zu einer Tiefe von  $n$  führt. Unter der Annahme sehr gleichmäßigen Wachstums hingegen existieren auf jeder Hierarchieebene des Segmentbaumes halb so viele Segmente wie auf der Ebene darunter, woraus eine Gesamttiefe von  $\log_2 n$  im optimalen Fall resultiert. Dieser Wert

kann nur erreicht werden, wenn  $n$  ein Vielfaches von 2 ist. Zusammengefasst gilt also

$$\log_2 n \leq d \leq n. \quad (2.16)$$

Zurück zur Implementierung des Suchverfahrens: Wenn zu einem Segment kein gegenseitig bestpassender Nachbar für eine Verbindung gefunden werden kann, so wird dieses Segment aus der Liste der Ausgangssegmente entfernt und das Attribut **final** auf den Wert **wahr** gesetzt. Bei einer späteren Segmentverbindung in der Nachbarschaft eines solchen Objekts, wird dieses Segment erneut in die Liste der Ausgangssegmente aufgenommen und das Attribut **final** zurück auf **falsch** gesetzt, weil sich dadurch die Werte für das Heterogenitätsmaß ändern und somit eine Heterogenität unterhalb des Schwellenwertes erneut möglich ist.

Findet der Algorithmus kein Segment mehr in der Liste der Ausgangssegmente, so bricht er ab und betrachtet die Segmentation als endgültig. Wie ein Testfall gezeigt hat, ergibt auch ein erneuter Start des Verfahrens durch Einfügen aller Segmente der obersten Hierarchieebene in die Liste der Ausgangssegmente keine weiteren Fusionen von Segmentpaaren.

Nach der Wahl des Ausgangssegments kann innerhalb jedes Iterationsschrittes des Algorithmus die Suche nach dem bestpassenden Nachbarn mithilfe des *Local Mutual Best Fitting* durchgeführt werden. Dazu liefert die Methode **findNearestNeighbour()** der Klasse **Segment** ein Objekt des Typs **PotentialMergeInformation** zurück, wenn das Ausgangssegment **seed** einen bestpassenden Nachbarn **mergescandidate** besitzt, der nicht als final markiert wurde und bei dem eine Fusion zudem den Schwellenwert für die Heterogenität **diffheterogeneitythres** nicht überschreitet. Ist zudem für **mergescandidate** der bestpassende Nachbar das Ausgangssegment **seed**, so werden die beiden Segmente verbunden. Andernfalls startet die Suche erneut mit **mergescandidate** als Ausgangssegment. Die Endlichkeit des Algorithmus ist auf diese Weise garantiert, da in jeder Iteration die Heterogenitätsdifferenz streng monoton fällt.

Werden auf die beschriebene Art zwei Segmente  $s_1$  und  $s_2$  gefunden, die zu einem neuen Segment  $s_3$  verbunden werden sollen, wird die Methode **merge()** der Klasse **Segment** aufgerufen. Darin werden die Werte für  $\hat{\mu}$  und  $\hat{\sigma}$  für  $s_3$  neu berechnet. Unter Zuhilfenahme der Wichtungsfaktoren

$$a_1 + a_2 = \frac{n_1}{n_1 + n_2} + \frac{n_2}{n_1 + n_2} = 1 \quad (2.17)$$

und bei Weglassung der Kanalindizes gilt

$$\begin{aligned} \hat{\mu}_3 &= a_1 \hat{\mu}_1 + a_2 \hat{\mu}_2, \\ \hat{\sigma}_3^2 &= a_1 \hat{\sigma}_1^2 + a_2 \hat{\sigma}_2^2 + a_1 a_2 (\hat{\mu}_1 - \hat{\mu}_2)^2. \end{aligned} \quad (2.18)$$

Denn, wie man leicht sieht, gilt für den empirischen Erwartungswert

$$\begin{aligned} \hat{\mu}_3 &= \frac{1}{n_3} \sum_{i=1}^{n_3} x_i \\ &= \frac{n_1}{n_1 + n_2} \frac{1}{n_1} \sum_{i=1}^{n_1} x_i + \frac{n_2}{n_1 + n_2} \frac{1}{n_2} \sum_{i=n_1+1}^{n_1+n_2} x_i \\ &= a_1 \hat{\mu}_1 + a_2 \hat{\mu}_2. \end{aligned} \quad (2.19)$$



Für die unkorrigierte, empirische Varianz gilt analog

$$\begin{aligned}
\hat{\sigma}_3^2 &= \frac{1}{n_3} \sum_{i=1}^{n_3} x_i^2 - \hat{\mu}_3^2 \\
&= \frac{1}{n_1 + n_2} \sum_{i=1}^{n_1+n_2} x_i^2 - (a_1 \hat{\mu}_1 + a_2 \hat{\mu}_2)^2 \\
&= \frac{1}{n_1 + n_2} \sum_{i=1}^{n_1+n_2} x_i^2 - a_1^2 \hat{\mu}_1^2 - 2a_1 a_2 \hat{\mu}_1 \hat{\mu}_2 - a_2^2 \hat{\mu}_2^2 \\
&= \frac{1}{n_1 + n_2} \sum_{i=1}^{n_1+n_2} x_i^2 - (a_1 - a_1 a_2) \hat{\mu}_1^2 - 2a_1 a_2 \hat{\mu}_1 \hat{\mu}_2 - (a_2 - a_1 a_2) \hat{\mu}_2^2 \\
&= \frac{a_1}{n_1} \sum_{i=1}^{n_1} x_i^2 - a_1 \hat{\mu}_1^2 + \frac{a_2}{n_2} \sum_{i=n_1+1}^{n_1+n_2} x_i^2 - a_2 \hat{\mu}_2^2 + a_1 a_2 (\hat{\mu}_1^2 - 2\hat{\mu}_1 \hat{\mu}_2 + \hat{\mu}_2^2) \\
&= a_1 \hat{\sigma}_1^2 + a_2 \hat{\sigma}_2^2 + a_1 a_2 (\hat{\mu}_1 - \hat{\mu}_2)^2.
\end{aligned} \tag{2.20}$$

Die eigentliche Farbheterogenität wird über Gleichung 2.8 berechnet.

Im Gegensatz zu einer Neuberechnung der Farbheterogenität aus den Originaldaten hat die Aktualisierungsmethode aus Gleichung 2.18 zwei entscheidende Vorteile. Erstens ist der Rechenaufwand konstant im Gegensatz zur vollständigen Neuberechnung mit einem Aufwand von  $\mathcal{O}(n)$ . Zweitens ist laut Chan u. a. (1979) auch der numerische Fehler geringer als bei einer vollständigen Neuberechnung.

Zur Neuberechnung der Formheterogenität wird ein Verfahren verwendet, das wie bei der Farbheterogenität die bereits vorhandenen Daten nutzt, um die Rechenzeit möglichst gering zu halten. Zunächst einmal werden die Ausmaße des kleinsten Rechtecks, das  $s_3$  beinhaltet, über die Gleichungen

$$\begin{aligned}
x_{\min}(s_3) &= \min \{x_{\min}(s_1), x_{\min}(s_2)\}, \\
y_{\min}(s_3) &= \min \{y_{\min}(s_1), y_{\min}(s_2)\}, \\
x_{\max}(s_3) &= \max \{x_{\max}(s_1), x_{\max}(s_2)\}, \\
y_{\max}(s_3) &= \min \{y_{\max}(s_1), y_{\max}(s_2)\},
\end{aligned} \tag{2.21}$$

berechnet. Anschließend ist es möglich, den Umfang des Rechtecks  $b$  aus diesen Werten zu ermitteln. Der Umfang des Segments  $s_3$  selbst wird über die Summe des Umfangs der Ausgangssegmente  $s_1$  und  $s_2$  abzüglich der zweifachen Länge der gemeinsamen Außenkante ermittelt, da diese durch die Verbindung der Segmente entfällt. Die Länge dieser gemeinsamen Außenkante  $l_{cb}(s_1, s_2)$  wird in den Objekten vom Typ `NeighbourhoodRelation` im Attribut `commonperimeter` gespeichert. Es gilt somit

$$l(s_3) = l(s_1) + l(s_2) - 2l_{cb}(s_1, s_2). \tag{2.22}$$

Die Fläche des neuen Segments kann offensichtlich sehr einfach über die Summe der Flächen der Ausgangssegmente bestimmt werden. Abschließend wird mithilfe der Gleichungen 2.12, 2.14 sowie 2.15 die Formheterogenität des neuen Segments ermittelt.

Neben der Neuberechnung der Heterogenität ist es ebenfalls wichtig, dass im Falle einer Verbindung für alle Nachbarn  $s_x$  von  $s_1$  und  $s_y$  von  $s_2$  in den `NeighbourhoodRelation`-Objekten

$N(s_1, s_x)$  und  $N(s_2, s_y)$  die Bezüge auf  $s_1$  und  $s_2$  durch das neue Segment  $s_3$  ersetzt werden. Dabei ist einerseits zu beachten, dass  $N(s_1, s_2)$  komplett gelöscht werden muss. Andererseits ist es im Falle der Nachbarschaft von  $s_1$  oder  $s_2$  zu einem beliebigen Segment  $s_x$  nötig, die Länge der gemeinsamen Begrenzungslinie durch

$$l_{cb}(s_3, s_x) = l_{cb}(s_1, s_x) + l_{cb}(s_2, s_x) \quad (2.23)$$

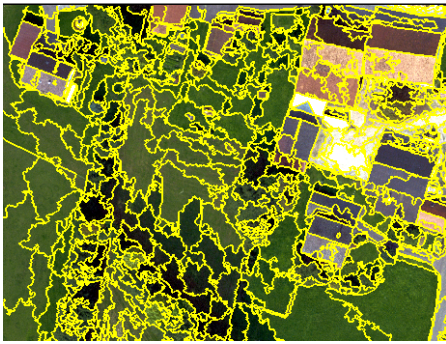
neu zu berechnen. Zusätzlich ist in diesem Fall entweder  $N(s_1, s_x)$  oder  $N(s_2, s_x)$  komplett zu löschen. Dadurch kann sichergestellt werden, dass bei der weiteren Suche nach zu verbindenden Segmenten sowohl keine doppelten Nachbarschaftsbeziehungen als auch keine Nachbarschaftsbeziehungen mit Beteiligung von Segmenten unterschiedlicher Hierarchieebenen existieren. Dass die Segmente  $s_1$  und  $s_2$  nicht komplett gelöscht werden, liegt darin begründet, dass bei der im nächsten Abschnitt beschriebenen Übertragung einer Segmentation auf andere Bilddaten der Entstehungsverlauf verwendet wird, um eine mit den neuen Daten konsistente Segmentierung zu erzeugen.

Die beschriebene Reimplementierung des *Multiresolution-Segmentation*-Algorithmus unterscheidet sich von der Originalimplementierung in eCognition im Wesentlichen nur in der fehlenden Nutzung von Dithermatrizen zur Bestimmung der Ausgangssegmente und der geringeren Effizienz in Bezug auf Laufzeit und Speicherbedarf (siehe dazu die sich anschließende Betrachtung zur Komplexität und Ressourcenbedarf). Bei der Betrachtung der Ergebnisse der beiden Verfahren finden sich bei gleichen Eingabeparametern jedoch kaum Unterschiede insbesondere bei anthropogenen Strukturen, wie in Abbildung 2.5 zu sehen ist. Aus diesem Grund ist es möglich, die vorgestellte Implementierung zur Nutzung im Rahmen der Änderungsdetektion zu erweitern. Dies ist der Hauptbestandteil des nächsten Abschnitts.

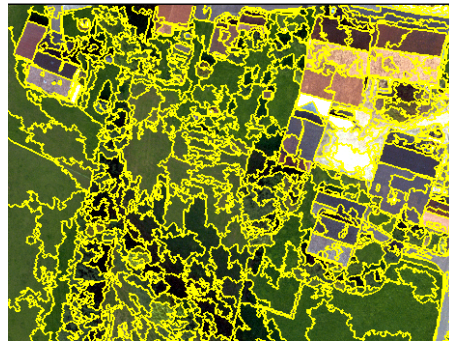
### Komplexität und Ressourcenbedarf

Die beschriebene Reimplementierung versucht, möglichst nah am Originalalgorithmus zu bleiben, um die in Abschnitt 2.3.3 beschriebenen Vorzüge ausnutzen zu können. Begründet durch die Tatsache, dass die Details des Algorithmus nicht vollständig veröffentlicht bzw. bewusst im Unklaren gelassen wurden (siehe Baatz u. a. 2003), kann im Rahmen dieser Arbeit nur versucht werden, eine effiziente Implementierung zu erhalten. An die Geschwindigkeit des Originalalgorithmus, der über ein Jahrzehnt hinweg optimiert wurde, kann diese jedoch nicht heranreichen. Im Folgenden soll nichtsdestoweniger eine Analyse sowohl der Komplexität des Algorithmus als auch der Schnelligkeit und des Speicherbedarfs der Reimplementierung im Vergleich zum Original durchgeführt werden.

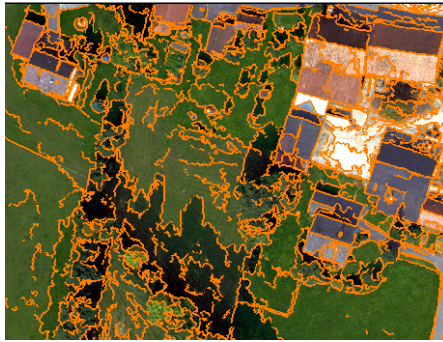
Die Implementierung des Verfahrens als Plugin befindet sich auf der dieser Arbeit beiliegenden CD im Verzeichnis **MRS4CD**. Die Eingabemaske mit den relevanten Parametern wird in Abbildung 2.18 dargestellt. Obwohl dieses Plugin für die Segmentierung im Rahmen der objekt-basierten Änderungsdetektion entwickelt wurde, welche in Abschnitt 2.3.4 beschrieben wird, kann es auch für die Einzelbildsegmentierung verwendet werden. Dazu sind alle **Map**-Parameter auf demselben Wert zu belassen und ansonsten nur die Parameter im Bereich **Segmentation of first image** zu beachten. In diesem Bereich wird über den Parameter **Level Name** der Name des zu erzeugenden Objektlayers festgelegt. In **Layerweights** 1 kann bestimmt werden, welche Kanäle des Bildes für die Segmentierung verwendet werden sollen und evtl. mit welcher Wichtung. Der **Scale parameter** entspricht dem Schwellenwert  $T$  in der obigen Beschreibung des Verfahrens. Über **Shape Weight** und **Compactness Weight** werden die Gewichte  $w_s$  und  $w_{comp}$  des Verfahrens spezifiziert. Schließlich kann mithilfe des Parameters **Mutual Tolerance** gesteuert werden, wie sehr das Prinzip des *Local Mutual Best Fitting* eingehalten werden soll. Bei einem Wert von 0 für diesen Parameter erfolgt eine strenge Befolgung dieses Paradigmas. Variiert man den Wert jedoch in Richtung von 1 geht das Verhalten



(a) Ergebnis des Originalalgorithmus.



(b) Ergebnis der reimplementierten Version.



(c) In beiden Varianten identisch extrahierte Objektgrenzen.

**Abb. 2.5.:** Vergleich der Ergebnisse des *Multiresolution-Segmentation*-Algorithmus der Originalimplementierung mit der reimplementierten Version.

des Algorithmus in die Methode des *Local Best Fitting* über.

Der *Multiresolution-Segmentation-Algorithmus* ist ein *Region-Merging*-Verfahren (siehe Abschnitt 2.3.2). Die rechnerische Komplexität dieser Algorithmen wurde vergleichend in Gofman (2006) analysiert. Es wurde darin allgemein gezeigt, dass eine direkte Implementierung die Komplexität  $\mathcal{O}(n^3)$  besitzt. Die dabei verwendete Landau-Notation, auch O-Notation genannt, bedeutet, dass für die Laufzeit bzw. für den Speicherbedarf  $f$  in Abhängigkeit von der Länge der Eingabe  $n$  ein  $n_0$  existiert, so dass für  $n > n_0$ , eine konstante  $c$  und die begrenzende Funktion  $g$  gilt

$$f(n) = \mathcal{O}(g(n)) \iff f(n) \leq cg(n). \quad (2.24)$$

Man beachte, dass das Gleichheitszeichen auf der linken Seite der Äquivalenz als „gerichtet“ betrachtet werden muss und insbesondere die Eigenschaft der Symmetrie nicht in jedem Fall gewährleistet ist.

Die weitere Betrachtung benötigt neben dieser formalen Definition auch einige Rechenregeln für die Landau-Notation. Eine erste wichtige Eigenschaft der Landau-Funktion  $\mathcal{O}$  ist ihre Additivität. Es gilt

$$\begin{aligned} \mathcal{O}(f(n)) + \mathcal{O}(g(n)) &= c_f f(n) + c_g g(n) \\ &\leq \max\{c_f, c_g\} (f(n) + g(n)) \\ &= \mathcal{O}(f(n) + g(n)), \end{aligned} \quad (2.25)$$

Eine zweite Summenregel, die häufig gebraucht wird, ergibt sich aus

$$\begin{aligned} \mathcal{O}(f(n) + g(n)) &= c_{fg}(f(n) + g(n)) \\ &\leq c_{fg} 2 \max\{f(n), g(n)\} \\ &= \mathcal{O}(\max\{f(n), g(n)\}). \end{aligned} \quad (2.26)$$

Als abschließende dritte wichtige Regel für die Verwendung der Landau-Notation existiert auch für Produkte eine Erhaltungsregel analog zu 2.25 mit

$$\begin{aligned} \mathcal{O}(f(n)g(n)) &= c_{fg}f(n)g(n) \\ &= c_f f(n)c_g g(n) \\ &= \mathcal{O}(f(n))\mathcal{O}(g(n)). \end{aligned} \quad (2.27)$$

Für die oben genannte Laufzeitkomplexität der *Region-Merging*-Algorithmen von  $\mathcal{O}(n^3)$  bedeutet diese Gleichung, dass die Laufzeitkomplexität  $C^{(r)}$  bis auf einen konstanten Faktor  $c$  in kubischer Abhängigkeit zur Eingabelänge  $n$  steht. D.h. es gilt

$$C_{\text{Region Merging}}^{(r)}(n) \leq cn^3. \quad (2.28)$$

In Gofman (2006) wird jedoch auch darauf hingewiesen, dass es von der Implementierung des jeweiligen *Region-Merging*-Verfahren abhängt, welche Komplexität der Algorithmus tatsächlich hat. Zudem sind neben der Eingabelänge  $n$  auch andere Einflussgrößen, wie etwa die resultierende Objektanzahl und die Kanalanzahl von Bedeutung und ermöglichen eine genauere Abschätzung. Für die vorgestellte Implementierung des *Multiresolution-Segmentation-Algorithmus* wird aus diesen Gründen im Folgenden eine Analyse der Komplexität präsentiert. Dazu seien  $n$  die Anzahl der Pixel des zu segmentierenden Bildes und  $k$  die Anzahl der Kanäle. Diese beiden Größen entsprechen der oben beschriebenen Länge der Eingabe. Für ein Bild der Breite  $a$  und der Höhe  $b$  gilt somit  $n = ab$ . Die Anzahl der Segmente in der endgültigen

Segmentation sei durch  $m$  gegeben.

In der Initialisierungsphase des Algorithmus wird für jedes der  $n$  Pixel ein Objekt der Klasse **Segment** mit jeweils vier Objekten vom Typ **NeighbourhoodRelation**, einem Objekt vom Typ **ColorProperties**,  $k$  Objekten vom Typ **ColorPropertiesPerChannel** und schließlich einem **ShapeProperties**-Objekt erstellt. Für diesen Teil der Initialisierung wird somit ein Rechenaufwand von  $c_1 kn + c_2$  Operationen benötigt. Anschließend fällt für das Einfügen in die Liste der Ausgangssegmente, **Seeds**, sowie die Herstellung der zufälligen Reihenfolge in dieser Liste ein Aufwand von  $c_3 n + c_4$  an (siehe *function template std::random\_shuffle*).

Zusammengefasst gilt für die Laufzeitkomplexität der Initialisierung des Algorithmus somit

$$C_{\text{init}}^{(r)} = c_1 nk + c_2 + c_3 n + c_4 = \mathcal{O}(kn). \quad (2.29)$$

Die rechnerische Komplexität des Hauptteils der *Multiresolution Segmentation* resultiert wesentlich aus der Anzahl der Fusionen  $f$ . Diese Anzahl ergibt sich aus der Differenz der Pixelanzahl  $n$  und der Objektanzahl  $m$  als

$$f = n - m. \quad (2.30)$$

Nachvollzogen werden kann das, indem man sich verdeutlicht, dass das Bild aus  $n$  Pixel besteht, die am Anfang die Ausgangssegmente stellen. Pro Fusion wird nun die Anzahl der Segmente um genau ein Segment reduziert bis nur noch  $m$  Segmente verbleiben. Daher werden für die Erzeugung der Segmentation  $n - m$  Fusionen benötigt.

Sei zu einem beliebigen Zeitpunkt die Anzahl der Segmente in der obersten Hierarchieebene, die vor einer Fusion existieren, gegeben durch  $i$  mit  $m \leq i \leq n$ . Dann fallen auch maximal  $i$  Aufrufe der Methode **findNearestNeighbour** in der Klasse **Segment** an. In dieser Methode wird abhängig von der Anzahl der Nachbarn und der Kanalanzahl die Heterogenität der hypothetisch entstehenden Segmente berechnet. Die Anzahl der Nachbarn eines jeden Segments ist durch  $i - 1$  beschränkt. Auch die Kanalanzahl  $k$  beeinflusst den Aufwand dieses Schrittes, weshalb eine Komplexität von

$$\begin{aligned} C_{\text{findNearestNeighbour}}^{(r)} &= \sum_{i=m}^n (c_5 ki(i-1) + c_6) \\ &= c_5 k \left( \sum_{i=1}^n i^2 - \sum_{i=1}^m i^2 + m^2 - \sum_{i=m}^n i \right) + c_6(n - m + 1) \\ &= c_5 k \left( \frac{n^3 - m^3}{3} + \frac{n^2 - m^2}{2} + \frac{n - m}{6} + m^2 - \frac{1}{2}(n - m + 1)(n + m) \right) \\ &\quad + c_6(n - m + 1) \\ &= c_5 k \left( \frac{n^3 - m^3}{3} + m^2 + \frac{5n - 7m}{6} \right) + c_6(n - m + 1) \\ &= \mathcal{O}(k(n^3 - m^2(m-1))) \end{aligned} \quad (2.31)$$

für das Finden des lokal gegenseitig bestpassenden Nachbarn entsteht.

In der Methode **merge**, welche die Fusion selbst bewerkstelligt, entsteht durch die Änderungen an den maximal  $i - 2$  Objekten vom Typ **NeighbourhoodRelation** einschließlich der Manipulation eines assoziativen Arrays, das Einfügen des Segments in die Liste der Ausgangssegmente

sowie weiteren, Operationen mit konstantem Aufwand eine Komplexität von

$$\begin{aligned}
 C_{\text{merge}}^{(r)} &= \sum_{i=m+1}^n (c_7(i-2) \log(i-2) + c_8) \\
 &\leq c_7 \log n \sum_{i=m+1}^n i + c_8(n-m) \\
 &= c_7 \log n \frac{1}{2}(n^2 - m^2 + n - m) + c_8(n-m) \\
 &= \mathcal{O}(\log n (n^2 - m^2)).
 \end{aligned} \tag{2.32}$$

Somit resultiert für den Hauptteil des Algorithmus eine Laufzeitkomplexität von

$$\begin{aligned}
 C_{\text{main}}^{(r)} &= C_{\text{findNearestNeighbour}}^{(r)} + C_{\text{merge}}^{(r)} \\
 &= \mathcal{O}(k(n^3 - m^2(m-1))).
 \end{aligned} \tag{2.33}$$

In einem abschließenden Schritt wird die Segmentkarte erstellt. Dafür wird für jedes der  $n$  Pixel in höchstens  $n-1$  Schritten der Segmentbaum durchlaufen, weshalb ein Aufwand von

$$C_{\text{final}}^{(r)} = c_9 n(n-1) + c_{10} = \mathcal{O}(n^2) \tag{2.34}$$

entsteht.

Für die Laufzeitkomplexität des *Multiresolution-Segmentation-Algorithmus* gilt somit

$$C_{\text{MRS}}^{(r)} = \mathcal{O}(k(n^3 - m^2(m-1))). \tag{2.35}$$

Wird in der Methode `findNearestNeighbour` eine sogenannte *Priority Queue* eingesetzt, so lässt sich dieser Aufwand auf  $\mathcal{O}(k(n^2 \log n - m(m-1) \log m))$  reduzieren (siehe Gofman 2006). Bei der im Rahmen dieser Arbeit erstellten Implementierung des Algorithmus wird darauf jedoch verzichtet, weil der Fokus auf der Anpassung für die objekt-basierte Änderungsdetektion liegen soll.

Neben den theoretischen Überlegungen zur Komplexität des Algorithmus ist jedoch auch die tatsächliche Laufzeit von Bedeutung, da über die Definition der O-Notation in Gleichung 2.24 nur eine obere Schranke beschrieben wird. Aus diesem Grund wird im Folgenden die Abhängigkeit der Laufzeit von den in der Komplexitätsbetrachtung verwendeten Parametern, d.h. der Eingabelänge  $n$ , der Kanalanzahl  $k$  und der Objektanzahl  $m$ , gemessen und analysiert. Dazu wird das bereits in Abbildung 2.5 verwendete Bild mit drei Kanälen und einer Ausdehnung von 890 mal 600 Pixeln genutzt. Zusätzlich kommt zur Betrachtung der Abhängigkeit zwischen Kanalanzahl und Laufzeit ein hyperspektraler Datensatz der Ausdehnung 249 mal 455 Pixeln und 125 Kanälen zum Einsatz. Die Rechnungen werden auf einem Computer mit einer CPU *Intel Xeon @ 2.5 GHz* und 16 GB RAM unter Verwendung von *Microsoft Windows 7 64 bit* durchgeführt.

Im Zusammenhang mit den Untersuchungen zur Laufzeit des Algorithmus wird zunächst die Beziehung zwischen dem Heterogenitätsschwellenwert  $T$  und der resultierenden Objektanzahl  $m$  untersucht. Dabei kann festgestellt werden, dass der in Schiewe u. a. (2001) aufgestellten Behauptung, diese Abhängigkeit sei linear, widersprochen werden muss. In Abbildung 2.6 wird der Zusammenhang graphisch in doppelt logarithmischer Skalierung dargestellt. Dort wird sehr deutlich ein linearer Zusammenhang der Form

$$\log m(\log T) = -a \log T + b' \tag{2.36}$$

mit  $b' = \log_{10} b > 0$  und  $a > 1$  sichtbar. Dies bedeutet aber, dass die Abhängigkeit zwischen beiden Größen von der Natur einer umgekehrten Proportionalität der Form

$$m(T) = \frac{b}{T^a} \quad (2.37)$$

ist.

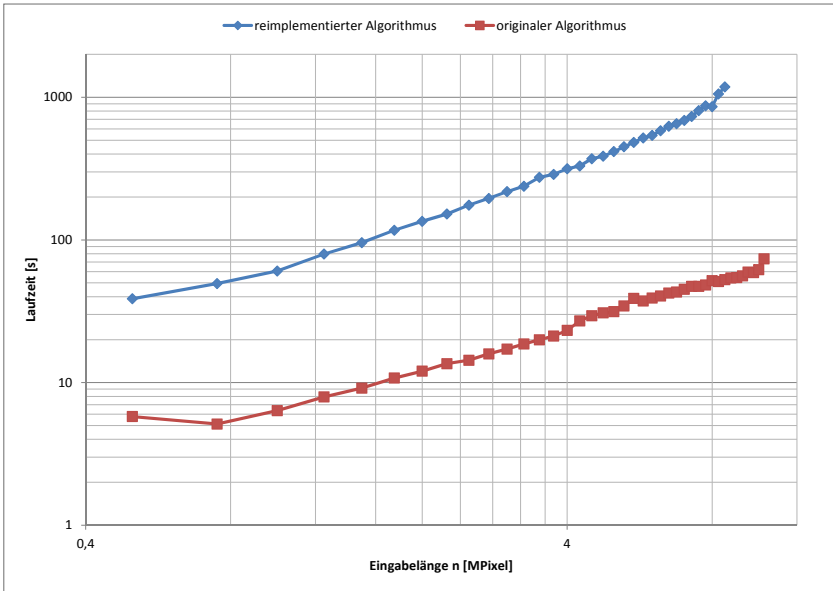


**Abb. 2.6.:** Abhängigkeit der resultierenden Objektanzahl  $m$  vom Heterogenitätsschwellenwert  $T$ .

Die wesentliche Einflussgröße für die Laufzeit des Algorithmus ist die Größe des Bildes, welche durch die Anzahl der vorhandenen Pixel  $n$  repräsentiert wird. In Abbildung 2.7 ist diese Abhängigkeit in doppelt logarithmischer Skalierung dargestellt und drei Punkte lassen sich daraus ableiten. Erstens wird gezeigt, dass die Annahme einer Komplexität von  $\mathcal{O}(n^3)$  durch die Daten nicht gerechtfertigt werden kann, da die Steigung beider Kurven nur unwesentlich größer als eins ist. Das bedeutet allerdings nicht, dass die Komplexitätsbetrachtung fehlerhaft ist, sondern nur, dass der betrachtete Parameterbereich noch nicht vom asymptotischen Verhalten des Algorithmus geprägt ist. Zweitens sieht man hier sehr deutlich, dass die Laufzeit der reimplementierten Version wie erwartet und eingangs besprochen in der Praxis um eine Größenordnung schlechter ist als die des Originalalgorithmus. Schließlich gilt drittens, dass für Bilder ab einer Größe von etwa fünf Megapixel die Laufzeit stärker ansteigt als zuvor. Der Grund dafür liegt in der Begrenzung der Hauptspeichermenge und der somit ab einer gewissen Bildgröße erfolgenden Auslagerung in den Festspeicher. Diesen Vorgang nennt man *Swapping*.

Für die weiteren Analysen der Einflussfaktoren auf die Laufzeit wird der Einfachheit halber nur die reimplementierte Version für die Komplexitätsanalyse betrachtet. Da die reimplementierte Version des Algorithmus im Fokus dieser Arbeit steht, erscheint dieser Schritt folgerichtig.

Neben der Eingabelänge  $n$  ist die Kanalanzahl  $k$  ein ebenfalls wichtiger Einflussfaktor auf die Laufzeit des Algorithmus. In Abbildung 2.8 wird diese Abhängigkeit dargestellt. Darin sieht man, dass die in Gleichung 2.35 auf theoretischen Argumenten beruhende Behauptung einer



**Abb. 2.7.:** Abhängigkeit der Laufzeit von der Eingabelänge  $n$ .

linearen Abhängigkeiten der Laufzeit von der Kanalanzahl auch mit den Laufzeitmessungen in der Praxis übereinstimmt. Die Datengrundlage für diese Betrachtung bildet eine hyperspektrale Aufnahme mit 125 Kanälen, wobei die Kanalanzahl schrittweise erhöht wurde.

Der dritte in Gleichung 2.35 erwähnte Parameter mit Einfluss auf die Laufzeit ist die Segmentanzahl  $m$ . In Abbildung 2.9 wird die Abhängigkeit der Laufzeit von diesem Faktor dargestellt. Die Abszisse wird dabei in logarithmischer Skalierung dargestellt. Aus der Abbildung ist ersichtlich, dass die in Gleichung 2.35 geäußerte Behauptung einer oberen Schranke kubischer Natur gerechtfertigt ist. Die Daten lassen sogar eine exponentielle Abhängigkeit plausibel erscheinen.

Über die Betrachtung der Laufzeit des Algorithmus hinaus ist auch der Bedarf an Hauptspeicher ein wesentlicher Aspekt in der Analyse der Leistungsfähigkeit des Verfahrens. Dazu ist es nötig, sowohl die Größe als auch die Maximalanzahl an Instanzen einer Klasse zu analysieren. Für eine solche Abschätzung werden zunächst einige Hilfsgrößen definiert. Die zum Zeitpunkt  $t$  im Verlauf des Algorithmus vorhandene Zahl der Segmente in der obersten Hierarchieebene sei  $m(t)$ . Für diese Größe gilt dabei offensichtlich eine Beschränkung durch die Anzahl der Pixel  $n$  als obere Schranke und die Anzahl der endgültigen Segmente  $m$  als untere Schranke, d.h.

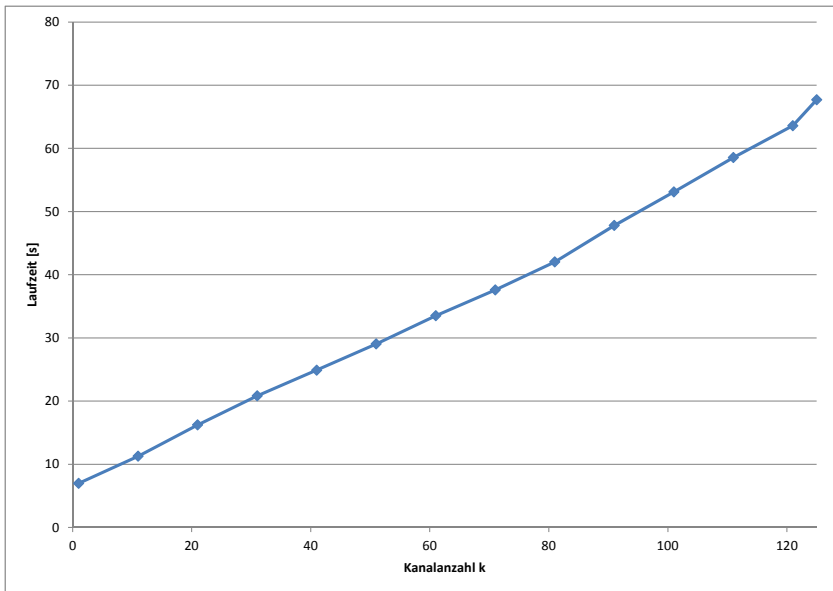
$$m \leq m(t) \leq n. \quad (2.38)$$

Des weiteren sei  $f(t)$  die Anzahl der Fusionen, die bis zum Zeitpunkt  $t$  stattgefunden haben. Da die Anzahl der Fusionen, wie im Zusammenhang mit 2.30 begründet wurde, gleich der Differenz zwischen der aktuellen Anzahl der Segmente in der obersten Hierarchieebene  $m(t)$  und der anfänglichen Segmentanzahl  $n$  ist gilt

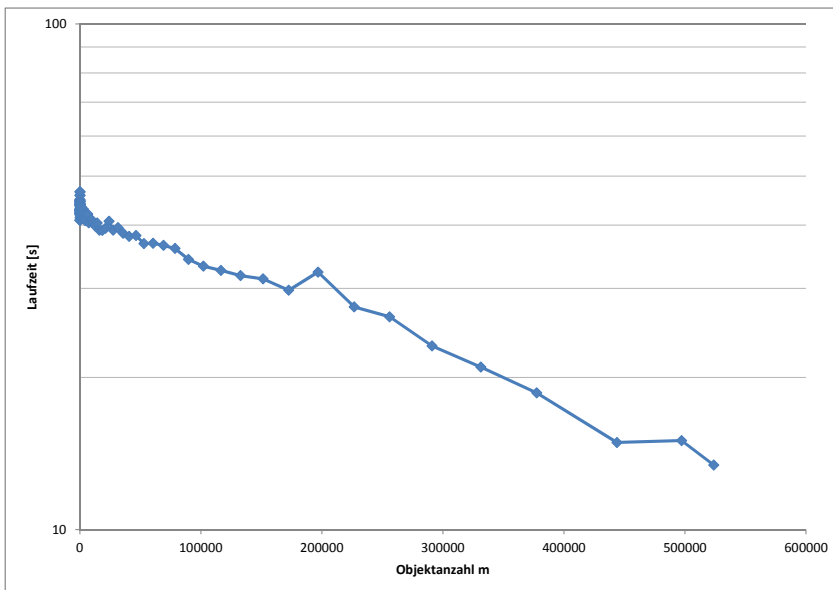
$$f(t) = n - m(t). \quad (2.39)$$

Schließlich wird im Folgenden die Anzahl der Hierarchieebenen benötigt. Diese Anzahl sei zum Zeitpunkt  $t$  durch  $d(t)$  definiert und ergibt sich aus der maximalen Tiefe aller Segmente. Bei völlig ungleichmäßigem Wachstum wird pro Fusion eine neue Hierarchieebene erzeugt,





**Abb. 2.8.:** Abhängigkeit der Laufzeit von der Kanalanzahl  $k$ .



**Abb. 2.9.:** Abhängigkeit der Laufzeit von der Objektanzahl  $m$ .

wodurch einschließlich der untersten Ebene  $n - m(t) + 1$  entstehen. Im Fall völlig gleichmäßigen Wachstums, d.h. wenn pro Hierarchieebene die Anzahl der Segmente halbiert wird, werden hingegen nur  $\log_2 n - \log_2 m + 1$  Ebenen benötigt. Somit gilt in Anlehnung an 2.16 für die Anzahl der Hierarchieebenen

$$\log_2 n - \log_2 m(t) + 1 \leq d(t) \leq n - m(t) + 1. \quad (2.40)$$

Für die eigentliche Größenabschätzung betrachte man zunächst die Instanzen der Klasse **NeighbourhoodRelation**. Die Größe dieser Objekte ist mit  $c_{11}$  konstant, da nur elementare Attribute bzw. Zeiger auf andere Klassen verwendet werden. Bei dieser Klasse muss jedoch für die Betrachtung der Anzahl zwischen jenen Instanzen, die Nachbarschaftsbeziehungen auf der Pixelebene repräsentieren, sogenannte *Low-Level Neighbourhood Relationships* (LLNHR), und jenen, die für Nachbarschaftsobjekte auf der höchsten Hierarchieebene stehen, sogenannte *Top-Level Neighbourhood Relationships* (TLNHR), unterschieden werden. Die Anzahl der LLNHR-Instanzen ist durch die Überlegung gegeben, dass horizontal in jeder der  $b$  Zeilen des Bildes  $a - 1$  und vertikal in jeder der  $a$  Spalten des Bildes  $b - 1$  LLNHR-Objekte vorhanden sind. Daraus resultiert

$$\begin{aligned} N_{LLNHR} &= b(a - 1) + a(b - 1) \\ &= 2ab - a - b \\ &= 2n - a - b. \end{aligned} \quad (2.41)$$

als Anzahl der LLNHR-Instanzen.

Die Anzahl der TLNHR-Instanzen kann durch die Überlegung, dass anfangs so viele TLNHR- wie LLNHR-Objekte existieren und durch jede Fusion diese Menge um zumindest eine Instanz reduziert wird, nach oben abgeschätzt werden durch

$$\begin{aligned} N_{TLNHR} &= 2n - a - b - f(t) \\ &= n - a - b + m(t). \end{aligned} \quad (2.42)$$

Insgesamt ist somit für die Klasse **NeighbourhoodRelation** die Speicherkomplexität mit

$$\begin{aligned} C_{\text{NeighbourhoodRelation}}^{(m)} &= c_{11}(N_{LLNHR} + N_{TLNHR}) \\ &= \mathcal{O}(n) + \mathcal{O}(n + m(t)) \\ &= \mathcal{O}(n) \end{aligned} \quad (2.43)$$

anzunehmen.

Für die Klasse **Segment** muss analog zur Klasse **NeighbourhoodRelation** unterschieden werden zwischen den Segmenten, die in der obersten Hierarchieebene liegen, sogenannten *Top-Level Segments* (TLS) und jenen Segmenten, die nicht in der obersten Hierarchieebene liegen, sogenannten *Non-Top-Level Segments* (NTLS). Bei den TLS-Instanzen ist die Größe mit  $c_{12}$  als konstant anzunehmen, da die Listen **lowlevelneighbours**, **mergecandidates** bei diesen Objekten leer sind und die Zeiger in **neighbours** bereits bei der Größenbetrachtung für TLNHR-Objekte erfasst wurden. Gemäß Definition sind  $m(t)$  TLS-Instanzen zum Zeitpunkt  $t$  vorhanden.

Die Instanzen des Typs NTLS besitzen einerseits dieselben konstanten Attribute der Größe  $c_{12}$  wie Objekte vom Typ TLS. Die Anzahl der NTLS-Objekte resultiert aus den  $n$  anfänglichen Segmenten zuzüglich genau einem Objekt pro Fusion abzüglich der Anzahl der TLS-Segmente. Es gibt somit

$$\begin{aligned} N_{NTLS} &= n + f(t) - m(t) \\ &= 2(n - m(t)) \end{aligned} \quad (2.44)$$

Instanzen der TLS-Segmente.

Auf der anderen Seite kommen jedoch zwei weitere Listen als Attribute hinzu. Die Größe der ersten Liste, `lowlevelneighbours`, kann für alle NTLS-Objekte gleichzeitig durch die Überlegung abgeschätzt werden, dass auf jeder der  $d(t)$  Hierarchieebenen ein LLNHR-Objekt höchstens zweimal in einem Segment referenziert wird, weshalb für die Anzahl dieser Zeiger gilt

$$\begin{aligned} N_{LLNHR*} &= 2(2n - a - b)f(t) \\ &= (4n - 2(a + b))(n - m(t)). \end{aligned} \quad (2.45)$$

Die Größe der einzelnen Zeiger kann mit  $c_{13}$  als konstant angenommen werden. Die Größe der zweiten Liste, `mergecandidates`, die Fusionskandidaten während des Prozesses speichert, kann mittels einer Summe durch alle Hierarchieebenen über die Anzahl der `NeighbourhoodRelation`-Instanzen für alle NTLS-Objekte gleichzeitig abgeschätzt werden. Es gilt somit für die Gesamtanzahl der Zeiger in den `mergecandidates`-Listen

$$\begin{aligned} N_{MC*} &= 2 \sum_{i=1}^{d(t)} (n - a - b + i) \\ &= 2 \frac{1}{2} d(t) (n - a - b + 1 + n - a - b + d(t)) \\ &= (n - m(t) + 1)(3n - 2a - 2b - m(t) + 2). \end{aligned} \quad (2.46)$$

Die Größe der einzelnen Zeiger ist mit  $c_{14}$  als konstant anzunehmen. Für die Klasse `Segment` ergibt sich somit eine Gesamtkomplexität von

$$\begin{aligned} C_{\text{Segment}}^{(m)} &= c_{12}N_{TLS} + c_{12}N_{NTLS} + c_{13}N_{LLNHR*} + c_{14}N_{MC*} \\ &= c_{12}m(t) + 2c_{12}(n - m(t)) + c_{13}(4n - 2(a + b))(n - m(t)) \\ &\quad + c_{14}(n - m(t) + 1)(3n - 2a - 2b - m(t) + 2) \\ &= \mathcal{O}(m(t)) + \mathcal{O}(n - m(t)) + \mathcal{O}(n(n - m(t))) + \mathcal{O}((n - m(t) + 1)(3n - m(t))) \\ &= \mathcal{O}(n^2). \end{aligned} \quad (2.47)$$

Die Anzahl der Instanzen der Klasse `PotentialMergeInformation` kann durch die Anzahl der TLNHR-Objekte mit

$$N_{\text{PotentialMergeInformation}} = n - a - b + m(t) \quad (2.48)$$

abgeschätzt werden. Da die Klasse nur elementare Attribute und Zeiger enthält, kann mit die Größe einer Instanz mit  $c_{15}$  als konstant angegeben werden. Deshalb erreicht die Gesamtkomplexität der Klasse `PotentialMergeInformation` die Größenordnung

$$\begin{aligned} C_{\text{PotentialMergeInformation}}^{(m)} &= c_{15}N_{\text{PotentialMergeInformation}} \\ &= \mathcal{O}(n - a - b + m(t)) \\ &= \mathcal{O}(n). \end{aligned} \quad (2.49)$$

Instanzen der Klasse `ShapeProperties` sind Subobjekte entweder der Klasse `PotentialMergeInformation` oder des Subtyps `TLS` der Klasse `Segment`, weshalb für die Anzahl der

Instanzen dieser Klasse die Abschätzung

$$N_{\text{ShapeProperties}} = n - a - b + 2m(t) \quad (2.50)$$

gilt. Die Größe einer Instanz ist mit  $c_{16}$  als konstant anzunehmen, woraus eine Gesamtkomplexität der Klasse **ShapeProperties** von

$$\begin{aligned} C_{\text{ShapeProperties}}^{(m)} &= c_{16} N_{\text{ShapeProperties}} \\ &= \mathcal{O}(n - a - b + m(t)) \\ &= \mathcal{O}(n). \end{aligned} \quad (2.51)$$

resultiert.

Für die Instanzenanzahl der Klasse **ColorProperties** gilt dieselbe Abschätzung wie für die Klasse **ShapeProperties**. Bei der Größe einer Instanz kommt jedoch zu einem konstanten Teil  $c_{17}$  aufgrund der Liste **channelinfo**, die Zeiger auf Objekte der Klasse **ColorPropertiesPerChannel** enthält und deren Länge abhängig von der Kanalanzahl ist, ein variabler Teil  $kc_{18}$  hinzu. Insgesamt ergibt sich somit eine Speicherplatzkomplexität von

$$\begin{aligned} C_{\text{ColorProperties}}^{(m)} &= (c_{17} + kc_{18}) N_{\text{ColorProperties}} \\ &= \mathcal{O}(k(n - a - b + 2m(t))) \\ &= \mathcal{O}(kn) \end{aligned} \quad (2.52)$$

für die Klasse **ColorProperties**.

Die bereits erwähnte Klasse **ColorPropertiesPerChannel** besitzt durch ihre elementaren Attribute eine konstante Größe  $c_{19}$  und wird  $k$  mal je Objekt der Klasse **ColorProperties** instanziiert. Daraus resultiert eine Gesamtkomplexität der Klasse **ColorPropertiesPerChannel** von

$$\begin{aligned} C_{\text{ColorPropertiesPerChannel}}^{(m)} &= c_{19} N_{\text{ColorPropertiesPerChannel}} \\ &= c_{19} k N_{\text{ColorProperties}} \\ &= \mathcal{O}(kn). \end{aligned} \quad (2.53)$$

Obwohl die Klasse **Pixel** in Abbildung 2.4 nur als Platzhalter für die Bilddaten dient, muss der Speicherbedarf abgeschätzt werden. Dazu wird angenommen, dass ein Wert eines der  $n$  Pixel in einem der  $k$  Kanäle einen konstanten Speicherplatzbedarf in Höhe von  $c_{20}$  besitzt. Somit ergibt sich eine Speicherkomplexität der Klasse **Pixel** von

$$\begin{aligned} C_{\text{Pixel}}^{(m)} &= c_{20} N_{\text{Pixel}} \\ &= c_{20} kn \\ &= \mathcal{O}(kn). \end{aligned} \quad (2.54)$$

Von der Klasse **Segmentation** existiert zu jedem Zeitpunkt im Verlauf des Algorithmus höchstens eine Instanz. Jedoch ist die Instanzgröße vom Status des Algorithmus und weiteren Faktoren abhängig. Der Speicherbedarf der elementaren Attribute sowie der Zeiger auf Objekte konstanter Größe werden mit  $c_{21}$  als konstant abgeschätzt. Die Liste **Segments** enthält alle existierenden Instanzen der Klasse **Segment**. Daher kann ihre Größe als  $c_{22}(2n - m(t))$  modelliert werden. Für die Größe der Liste **Seeds**, welche alle TLS enthält, die nicht als final markiert

sind, gilt eine Abschätzung von  $c_{23}m(t)$ . Die Liste `LowLevelNHRs2BeDeleted` kann maximal so viele Elemente enthalten, wie es LLNHR-Instanzen gibt. Daraus resultiert ein Betrag von  $c_{24}(2n - a - b)$  zum Speicherbedarf. Eine weitere Datenstruktur in der Klasse `Segmentation`, deren Speicherbedarf nicht konstant ist, findet sich im Attribut `layerWeights`. Die Länge dieser Liste ist direkt von der Kanalanzahl  $k$  abhängig, woraus sich ein Platzbedarf von  $c_{25}k$  ergibt. Schließlich existiert in der Klasse `Segmentation` das Attribut `segmentMap`, welches ein Array mit  $n$  Einträgen beinhaltet. Für dieses Array ergibt sich ein Speicherplatzbedarf von  $c_{26}n$ . In Summe wird für die Klasse `Segmentation` ein Platz im Speicher der Ordnung

$$\begin{aligned} C_{\text{Segmentation}}^{(m)} &= c_{21} + c_{22}(2n - m(t)) + c_{23}m(t) + c_{24}(2n - a - b) + c_{25}k + c_{26}n \\ &= \mathcal{O}(1) + \mathcal{O}(n - m(t)) + \mathcal{O}(m(t)) + \mathcal{O}(n) + \mathcal{O}(k) + \mathcal{O}(n) = \mathcal{O}(n + k) \end{aligned} \quad (2.55)$$

benötigt.

Insgesamt ergibt sich für den *Multiresolution-Segmentation*-Algorithmus somit eine Speicherkomplexität von

$$\begin{aligned} C_{\text{MRS}}^{(m)} &= C_{\text{Neighbourhoodrelation}}^{(m)} + C_{\text{Segment}}^{(m)} + C_{\text{PotentialMergeInformation}}^{(m)} + C_{\text{ShapeProperties}}^{(m)} \\ &\quad + C_{\text{ColorProperties}}^{(m)} + C_{\text{ColorPropertiesPerChannel}}^{(m)} + C_{\text{Pixel}}^{(m)} + C_{\text{Segmentation}}^{(m)} \\ &= \mathcal{O}(n) + \mathcal{O}(n^2) + \mathcal{O}(n) + \mathcal{O}(n) + \mathcal{O}(kn) + \mathcal{O}(kn) + \mathcal{O}(kn) + \mathcal{O}(n + k) \\ &= \mathcal{O}(n(n + k)). \end{aligned} \quad (2.56)$$

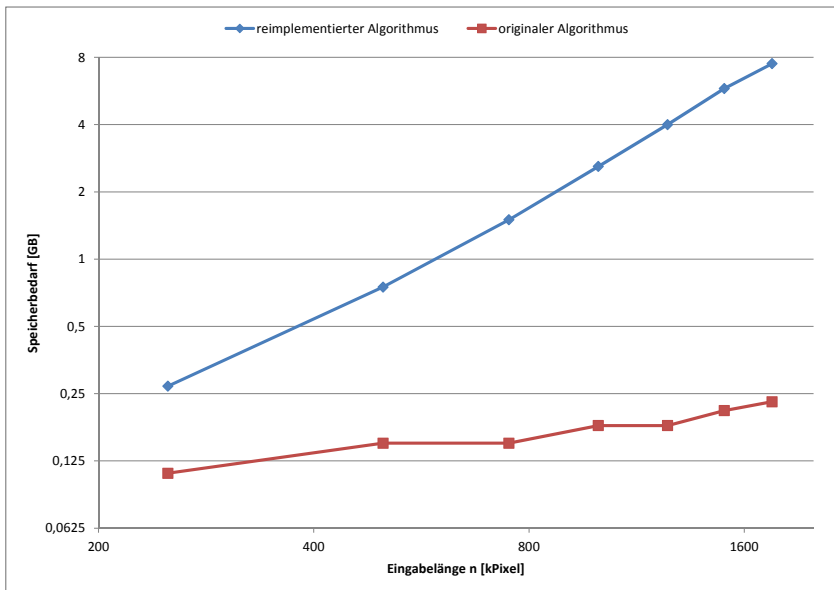
Um diese Abschätzung zu überprüfen, wurde ein iterativ vergrößerter quadratischer Ausschnitt eines Satellitenbildes mithilfe des Algorithmus segmentiert. Die Parameter blieben dabei über alle Messungen gleich. In Abbildung 2.10 sind die Ergebnisse der Messung graphisch dargestellt. Es zeigt sich, dass die Abschätzung der Speicherkomplexität in 2.56 mit den Ergebnissen der Messung übereinstimmt, da der Anstieg der Gerade rund zwei beträgt. Eine erweiterte Messung des Speicherbedarfs, bei der auch die Kanalanzahl variiert wird, könnte analog durchgeführt werden.

Ein zweiter Aspekt, der in Abbildung 2.10 sichtbar wird, ist die Tatsache, dass die reimplementierte Version des Algorithmus einen wesentlich höheren Speicherbedarf aufweist als die Originalversion. Der Grund für dieses Verhalten findet sich im Verhalten des Algorithmus bei der Fusion zweier Segmente. Im Fall einer Segmentierung eines einzelnen Bildes können nämlich die bestehenden Segmente gelöscht werden. Lediglich die Referenzen auf die Pixel des Bildes müssen gespeichert übernommen werden. Bei der reimplementierten Version des Algorithmus wird jedoch für den Einsatz im Rahmen der Änderungsdetektion die Entstehungsgeschichte eines Segments benötigt. Daher können die Ausgangssegmente der Fusion nicht gelöscht werden und es kommt zu dem beobachteten Mehrbedarf an Speicher.

### 2.3.4. Bildsegmentierung für die objekt-basierte Änderungsdetektion

#### Problembeschreibung

Der in Abschnitt 2.3.3 erläuterte Algorithmus der *Multiresolution Segmentation* kann für die objekt-basierte Änderungsdetektion prinzipiell direkt auf drei verschiedene Arten eingesetzt werden. Die erste Variante segmentiert die Bilddaten beider Aufnahmezeitpunkte allein auf Grundlage der Daten des ersten Aufnahmezeitpunkts. Dies entspricht der Segmentierung des ersten Bildes und der Übernahme dieser Segmentierung auf das zweite Bild. Der Vorteil dieser Art der Segmentierung liegt in der Tatsache, dass nur eine Segmentierung erstellt wird, die

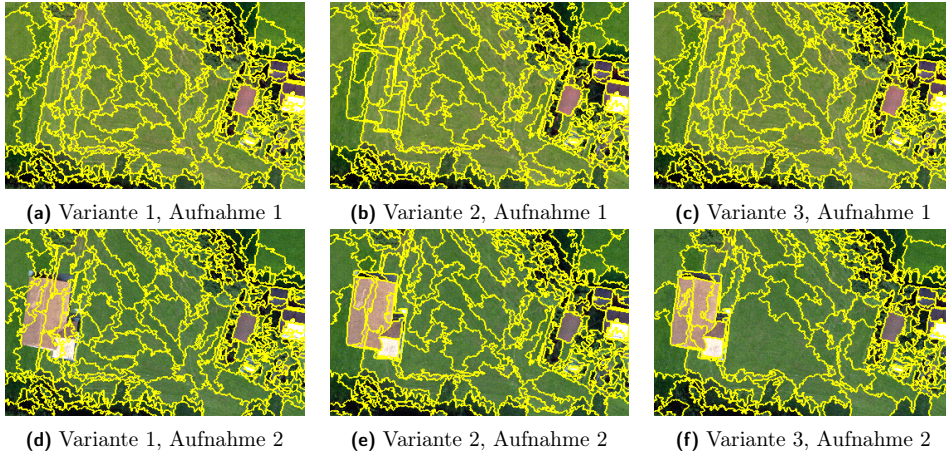


**Abb. 2.10.:** Abhängigkeit des Speicherbedarfs von der Eingabelänge  $n$ .

für die weitere Verarbeitung beider Bilder verwendet werden kann. Die Formeigenschaften der Objekte können für eine Analyse der Änderungen dadurch jedoch nicht herangezogen werden, da keine Änderung der Form innerhalb einer Segmentierung abgebildet werden kann. Außerdem hat dieser Ansatz zur Folge, dass die Bilddaten des zweiten Aufnahmezeitpunktes bei Änderungen der Form der Bildobjekte nicht korrekt segmentiert werden. Dieser Nachteil kann in einem zweiten Ansatz gemildert werden, indem die Segmentierung nicht nur auf den Kanälen des ersten Bildes, sondern auf der Grundlage aller Bildkanäle des ersten und des zweiten Bildes gemeinsam durchgeführt wird. Auch bei dieser Herangehensweise ist ein Vorteil für die weitere Verarbeitung, dass nur eine Segmentation erzeugt wird. Jedoch können hierbei erneut die Formeigenschaften nicht für die Änderungsdetektion genutzt werden. Durch die gleichzeitige Verwendung von Bilddaten unterschiedlicher Aufnahmezeitpunkte kann zudem nicht gewährleistet werden, dass die Segmentation die Bildobjekte in beiden Aufnahmen im Fall von Formänderungen ausreichend genau wiedergibt. Die dritte Variante schließlich ist die unabhängige Segmentierung beider Bilder, was als Konsequenz einerseits eine adäquate Segmentation der zwei Aufnahmen erzeugt. Andererseits können die Ergebnisse der Segmentierung nicht direkt im weiteren Prozess verarbeitet werden, da die zwei Segmentationen zunächst verbunden werden müssen. Der zweite und wesentliche Nachteil dieses Ansatzes besteht hingegen darin, dass durch die geringe Stabilität der existierenden Segmentierungsalgorithmen bereits lokale Änderungen zwischen den Aufnahmezeitpunkten zu globalen Änderungen der Segmentation führen (siehe dazu auch Abschnitt 2.3.1). Das wiederum wirkt sich auf die Stabilität der Werte aller Segmenteigenschaften aus. Daher kann es bei Verwendung dieser Variante allein durch die Segmentierung zur Detektion von Änderungen in Bereichen kommen, in denen keine oder nur geringe Änderungen stattgefunden haben.

Alle drei Varianten wurden bereits in Niemeyer (2001) erwähnt. Zur Anschauung sind die Segmentierungsansätze in Abbildung 2.11 dargestellt. Aufgrund der genannten Nachteile wird in diesem Abschnitt eine vierte Variante der Segmentierung vorgestellt. Dieser Ansatz erweitert den in Abschnitt 2.3.3 erläuterten *Multiresolution-Segmentation-Algorithmus*, so dass

die Ergebnisse einerseits auf beide Bilder angepasst sind und andererseits in Bereichen ohne Formänderung über beide Aufnahmen hinweg stabil bleiben. Zudem lässt sich der erweiterte Algorithmus in den Ablauf der objekt-basierten Änderungsdetektion integrieren.



**Abb. 2.11.:** Originalvarianten der Segmentierung für die objekt-basierte Änderungsdetektion mit Parametern  $T = 25$ ,  $w_s = 0.1$  und  $w_{comp} = 0.5$ .

In der Literatur existieren zum Problem der stabilen Segmentierung von Bildfolgen wenige Arbeiten. Zu nennen sind jedoch Beiträge, die sich mit der Segmentierung von Video beschäftigen, z.B. in Donoser u. a. (2011). Diese Ansätze unterscheiden sich jedoch vom hier vorliegenden Problem, da mehr als zwei Aufnahmezeitpunkte mit geringeren Zeitunterschieden und somit meist auch weniger starken Änderungen zwischen den Bildern verarbeitet werden.

### Ablauf des Verfahrens

Das im Rahmen dieser Arbeit entwickelte Verfahren zur Segmentierung für die objekt-basierte Änderungsdetektion basiert auf einer relativ einfachen Idee. Zunächst wird mithilfe des MRS-Algorithmus das Bild des ersten Aufnahmezeitpunkts segmentiert. Anschließend kann diese Segmentation auf das Bild des zweiten Aufnahmezeitpunktes übertragen werden. Dabei wird für jedes Segment mittels eines Konsistenztests (siehe S.73f) geprüft, ob es auch auf Grundlage der neuen Daten in sich konsistent ist. Sollte dieser Test fehlschlagen, wird das Segment für die Segmentation des zweiten Bildes mithilfe eines Verfahrens zur Segmentbereinigung (siehe S.74ff) aufgeteilt. Die so gewonnenen Segmentation wird in einem abschließenden Schritt als initiale Segmentation für den *Multiresolution-Segmentation-Algorithmus* genutzt. Dieser erstellt damit eine endgültige Segmentation des zweiten Bildes, welche sich nur in jenen Bereichen unterscheidet, wo entweder ein inkonsistentes Segment aufgeteilt wird oder eine über die erste Segmentation hinausgehende Fusion benachbarter Segmente möglich ist. Da dieses Verfahren den *Multiresolution-Segmentation-Algorithmus* für die Anwendung im Rahmen der objekt-basierten Änderungsdetektion erweitert, soll es *Multiresolution Segmentation for Change Detection* (MRS4CD) genannt werden. Eine detailliertere Vorschrift des Verfahrens findet sich in Tabelle 2.1.

Zwei Aspekte des Verfahrens bedürfen einer näheren Betrachtung. Erstens muss geklärt werden, wie die Konsistenz der Fusionen mit den neuen Bilddaten überprüft werden kann. Zweitens ist zu beschreiben, auf welche Art inkonsistente Segmente aufgeteilt werden können. Beide Fragen sollen im Folgenden beantwortet werden und die Implementierung ihrer Lösung

Schritt	Beschreibung
1	Segmentiere das Bild $I_1$ mithilfe des <i>Multiresolution-Segmentation-Algorithmus</i> .
2	Übertrage die so erhaltene Segmentation auf das Bild $I_2$ .
3	Überprüfe jede Fusion der Segmentation, d.h. jedes Segment bestehend aus mehr als einem Pixel, auf Konsistenz mit $I_2$ unter Verwendung eines Testkriteriums.
4	Entferne alle inkonsistenten Segmente aus der Segmentation mithilfe einer Prozedur zur Segmentbereinigung.
5	Führe auf der so erhaltenen Grundlage erneut den <i>Multiresolution-Segmentation-Algorithmus</i> aus. Dabei sei die Menge der Kandidatensegmente beschränkt auf die Bereiche, in denen Aufteilungen der Segmente stattgefunden haben.
6	Abschließend wird der <i>Multiresolution-Segmentation-Algorithmus</i> erneut ohne die Beschränkung der Kandidatensegmente ausgeführt, um eine endgültige auf $I_2$ angepasste Segmentation zu erhalten.

**Tab. 2.1.:** Verfahrensschritte des Segmentierungsalgorithmus *Multiresolution Segmentation for Change Detection*.

anhand des in Abschnitt 2.3.3 vorgestellten Objektmodells sowie von Beispielen erläutert werden.

### Methoden zur Konsistenzüberprüfung

Für die Problemstellung der Konsistenzüberprüfung wurden im Rahmen dieser Arbeit aufbauend auf Listner (2008) die vier folgenden Methoden entwickelt und implementiert:

- *Threshold Test*,
- *Relative Threshold Test*,
- *Local Best Fitting Test* und
- *Local Mutual Best Fitting Test*.

Im vorliegenden Abschnitt werden diese Testmethoden erläutert.

Der erste Test, genannt *Threshold Test*, überprüft für ein Segment  $s_3$ , das durch Fusion der Segmente  $s_1$  und  $s_2$  entstanden ist, ob der Heterogenitätsverlust bei der Fusion unterhalb eines Schwellenwertes  $T_t$  liegt, d.h. ob gilt

$$\Delta h(s_1, s_2) < T_t. \quad (2.57)$$

Diese einfachste aller Methoden zur Konsistenzüberprüfung untersucht nur, ob die Minimalbedingung aus Gleichung 2.5 auch für die Daten des zweiten Aufnahmezeitpunkts gilt. Inwiefern die Fusion optimal ist, spielt hierbei keine Rolle. Die Wahl des Schwellenwertes  $T_t$  sollte ungefähr im Bereich des für die Fusionen im ersten Schritt genutzten Schwellenwertes  $T$  liegen.

Ein zweiter auf dem Schwellenwerttest aufbauender Weg der Konsistenzüberprüfung nennt sich *Relative Threshold Test*. Diese Methode betrachtet die Differenz zwischen Heterogenitätsverlust bei der Fusion von  $s_1$  und  $s_2$  auf Grundlage der Daten des zweiten Aufnahmezeitpunkts  $\Delta h'(s_1, s_2)$  und dem Heterogenitätsverlust bei den Daten des ersten Aufnahmezeitpunkts



$\Delta h(s_1, s_2)$  im Verhältnis zum Heterogenitätsverlust bei den Daten des ersten Aufnahmezeitpunkts. Für dieses Verhältnis wird geprüft, ob es unterhalb eines Schwellenwertes  $T_{rt}$  liegt, d.h. ob gilt

$$\frac{\Delta h'(s_1, s_2) - \Delta h(s_1, s_2)}{\Delta h(s_1, s_2)} = \frac{\Delta h'(s_1, s_2)}{\Delta h(s_1, s_2)} - 1 < T_{rt}. \quad (2.58)$$

Die Werteskala für den Schwellenwert  $T_{rt}$  liegt im Prozentbereich, wobei auch Werte weit über 100% durchaus im Rahmen der üblichen Parametrisierung liegen. Insbesondere auf den niedrigen Hierarchieebenen kann es vorkommen, dass  $\Delta h(s_1, s_2)$  sehr klein wird und selbst bei geringen Änderungen ein großer Wert für  $T_{rt}$  gewählt werden muss, damit keine Aufteilung der Segmente erfolgt. Aus diesem Grund kann man die Überlegung anstellen, das Verfahren erst ab einer bestimmten Hierarchieebene zu beginnen. In der Implementierung für *eCognition* wird dies mittels des Parameters **Minimum Tree Depth** realisiert. Auch der *Relative Threshold Test* benötigt eine Speicherung der Segmentstruktur über alle Ebenen, damit die Prüfung der Fusionen auf Grundlage der neuen Daten möglich ist.

Die dritte Methode zur Konsistenzüberprüfung untersucht, ob der bei einer Fusion gewählte Nachbar auch unter den neuen Daten der lokal bestpassende ist. Zusätzlich wird weiterhin auch der Schwellenwert  $T_l$  als Kriterium wie beim *Threshold Test* genutzt. Konkret wird bei einem Segment  $s_3$ , welches aus der Fusion von  $s_1$  und  $s_2$  hervorgegangen ist, für  $s_1$  überprüft inwiefern  $s_2$  der lokal bestpassende Nachbar in  $I_2$  ist. Dieser Test wird *Local Best Fitting Test* genannt. Der Ansatz benötigt die Liste aller bei der Fusion berücksichtigter Nachbarsegmente damit die Prüfung auf Grundlage der Daten des zweiten Aufnahmezeitpunkts nachvollzogen werden kann. In der Implementierung wird dies über das Attribut **mergescandidates** in der Klasse **Segment** realisiert.

Eine direkte Erweiterung des *Local Best Fitting Tests* auf die Überprüfung, ob umgekehrt auch für  $s_2$  das Segment  $s_1$  der bestpassende Nachbar unter Berücksichtigung der Daten des zweiten Aufnahmezeitpunktes ist, ergibt den vierten Test. Dieser Test wird *Local Mutual Best Fitting Test* genannt.

Für beide Tests kann mittels des in Abschnitt 2.3.3 erläuterten Parameters **Mutual Tolerance** eingestellt werden, wie groß die Übereinstimmung zwischen den Daten des ersten und des zweiten Aufnahmezeitpunkts in Bezug auf die bestpassenden Segmente sein muss, damit ein Segment nicht als inkonsistent gilt.

Alle vier Verfahren markieren jene Segmente der Segmentation auf Grundlage der ersten Aufnahme, welche nicht mit den Daten des zweiten Aufnahmezeitpunktes konsistent sind. Es werden dabei nicht nur Segmente der obersten Hierarchieebene überprüft, sondern des gesamten binären Segmentbaumes.

### Aufteilung inkonsistenter Segmente

Ausgehend von den Ergebnissen der Konsistenzüberprüfung sollen im Folgenden drei verschiedene Varianten aufgezeigt werden, die das Problem der Aufteilung inkonsistenter Segmente lösen. Diese Methoden werden als

- *Universal Removal Strategy*,
- *Global Removal Strategy* und
- *Local Removal Strategy*

bezeichnet.

Der erste Ansatz entspringt der direkten Überlegung, ein inkonsistentes Segment auf seine grundlegenden Bestandteile zurückzuführen. Diese Herangehensweise wird *Universal Removal Strategy* genannt. Dabei wird jedes Segment der obersten Hierarchieebene, in dessen Segmentbaum ein inkonsistentes Segment gefunden wurde, auf die elementaren Pixel reduziert. Alle



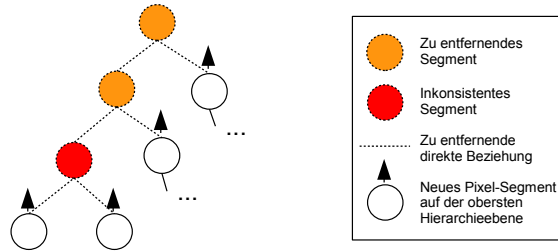


Abb. 2.13.: Segmentbereinigung mittels *Global Removal Strategy*.

gestellt.

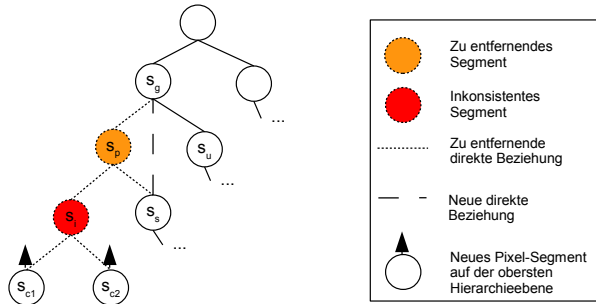


Abb. 2.14.: Segmentbereinigung mittels *Local Removal Strategy*.

Bei diesem Ansatz kann es jedoch vorkommen, dass die Segmente  $s_s$  und  $s_u$  nicht benachbart sind. In diesem Fall ist es nicht möglich, das Verfahren direkt auszuführen, weil sonst ein Segment  $s_g$  entstehen würde, dessen Kindsegmente nicht benachbart wären. Dies widerspricht der Definition eines Segments in 2.1. Stattdessen werden in Anlehnung an die *Global Removal Strategy* solange die Elternsegmente entfernt, bis ein Segment  $s'_i$  gefunden wird, dessen Geschwistersegment  $s'_s$  mit dem zugehörigen Segment  $s'_u$  benachbart ist, damit ein Segment  $s'_g$  entsteht, das der Definition 2.1 genügt. Die modifizierte *Local Removal Strategy* ist in 2.15 graphisch dargestellt.

Der Hauptvorteil der *Local Removal Strategy* liegt in der minimalen Modifikation des Segmentbaumes durch das Verfahren. Dadurch kann eine höhere Stabilität der Ergebnisse erreicht werden, d.h. der Unterschied zwischen der Segmentation von  $I_1$  und jener von  $I_2$  wird so gering wie möglich gehalten. Außerdem sind im Idealfall pro inkonsistentem Segment nur zwei Segmentobjekte sowie die zugehörigen Unterobjekte zu löschen. Diesem Vorteil steht jedoch der Nachteil gegenüber, dass die im Zusammenhang verwendete Methode zur Konsistenzüberprüfung sehr exakt sein muss, damit eine korrekte Segmentation von  $I_2$  erstellt werden kann. Zusätzlich ist zu bedenken, dass das neu zusammengestellte Segment  $s_g$  bzw.  $s'_g$  nicht notwendigerweise homogen sein muss, da die Pixel des Segments  $s_i$  nicht mehr Teil von  $s_g$  sind. Daher sollten  $s_g$  und deren Vorläufer im Anschluss auf ihre Konsistenz mit der neuen Struktur sowie den Bilddaten des zweiten Aufnahmezeitpunkts geprüft werden.

Für einen besseren Vergleich der verschiedenen Methoden zur Entfernung von inkonsistenten Segmenten werden in Abbildung 2.16 die Ergebnisse der drei erläuterten Verfahren beispielhaft dargestellt. Dazu kommen dieselben Daten zur Anwendung wie eingangs dieses Abschnitts in Abbildung 2.11. Sehr deutlich ist zu sehen, dass die Intensität der Änderungen an der Segmentation in Bereichen ohne wesentliche Änderungen der Bilddaten von links nach rechts

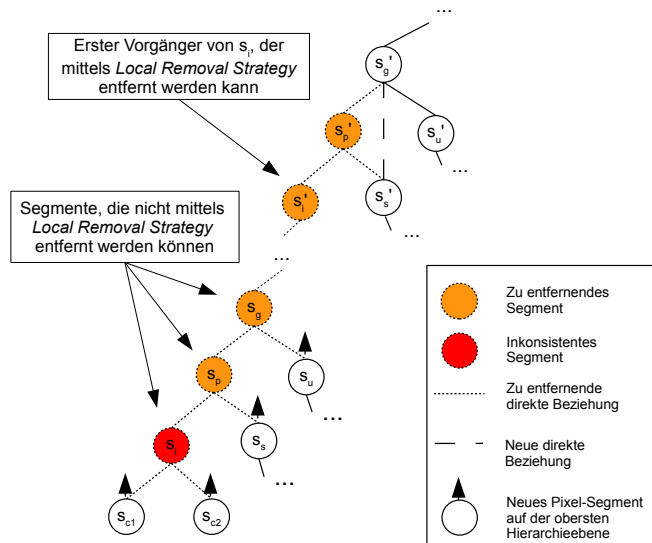


Abb. 2.15.: Segmentbereinigung mittels angepasster *Local Removal Strategy*.

abnimmt.

Ein Aspekt, der in der praktischen Verwendung des Verfahrens eine große Rolle spielt, ist die Frage nach der richtigen Wahl des **Scale Parameter** für die Methode zur Konsistenzüberprüfung. Die Objektvariable im Feld **maximum Tvalue** kann zu diesem Zweck genutzt werden. Diese ermöglicht es dem Nutzer, interaktiv den Wert für **Scale parameter** festzulegen. Dabei wird das Verfahren zunächst mit einem sehr großen Wert für **Scale parameter** im Parameterbereich **Consistency Test** ausgeführt, so dass kein Segment aus der ersten Segmentation aufgeteilt wird. Anschließend kann mittels der Funktionen **Feature View** und **Image Object Information** in *eCognition* der Schwellenwert interaktiv gewählt werden (siehe Abbildung 2.17). In einem erneuten Verfahrensdurchlauf kommt der so gewählte Schwellenwert dann zum Einsatz, um die gewünschte Aufteilung der inkonsistenten Segmente zu erreichen.

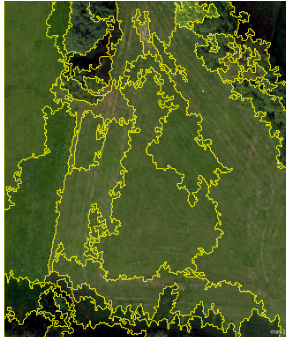
Eine alternative Möglichkeit, den Parameter zu wählen, ist die Nutzung von überwachten Methoden zur Schwellenwertbestimmung wie sie in Abschnitt 2.6.2 beschrieben werden. Dort wird auf diese Art der Parameterdefinition näher eingegangen.

### Anwendung des Plugins

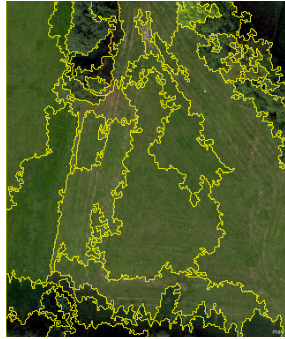
Der MRS4CD-Algorithmus wurde im Rahmen dieser Arbeit als Plugin für *eCognition* implementiert. Eine Kopie des Plugins befindet sich auf der beiliegenden CD im Verzeichnis **MRS4CD**. Dort befindet sich auch ein Beispiel für die Nutzung des Plugins im Unterverzeichnis **example**. Im Folgenden werden die Parameter des Plugins sowie einige Implementierungsdetails erläutert.

Für eine Übersicht der Eingabegrößen, die vom Benutzer an das Plugin übergeben werden müssen, sei zunächst auf Abbildung 2.18 verwiesen.

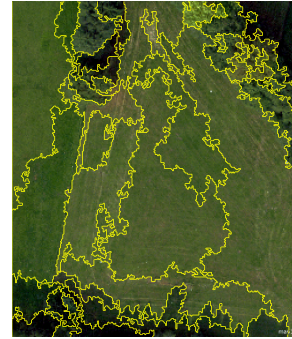
Die Eingabemaske ist in vier Bereiche unterteilt, die zu den verschiedenen in Tabelle 2.1 beschriebenen Abschnitten des Algorithmus korrespondieren. Jeder der vier Parameterbereiche besitzt einen eigenen **Map** sowie einen Parameter **Level Name**. Dadurch wird erreicht, dass nicht nur die endgültigen Segmentierungen als Ergebnis des Verfahrens zurückgeliefert werden, sondern auch alle Zwischenstände erhalten bleiben. Für die Wahl der richtigen Parameter



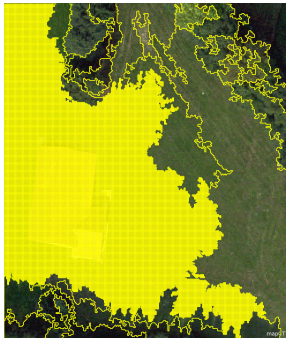
(a) Ergebnis Phase 1 mit *Universal Removal Strategy*.



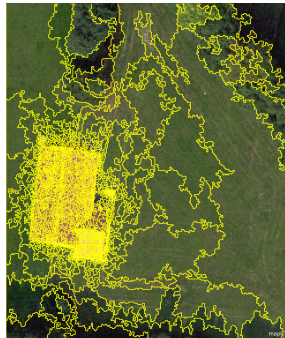
(b) Ergebnis Phase 1 mit *Global Removal Strategy*.



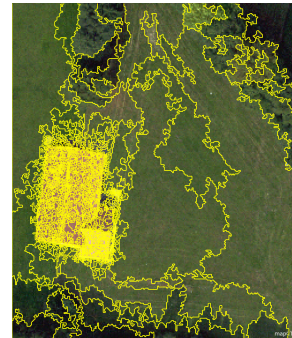
(c) Ergebnis Phase 1 mit *Local Removal Strategy*.



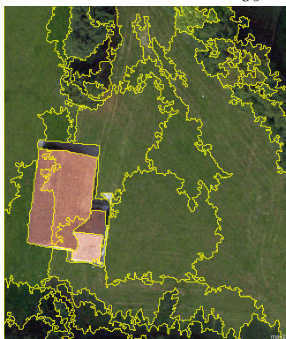
(d) Ergebnis Phasen 2-4 mit *Universal Removal Strategy*.



(e) Ergebnis Phase 2-4 mit *Global Removal Strategy*.



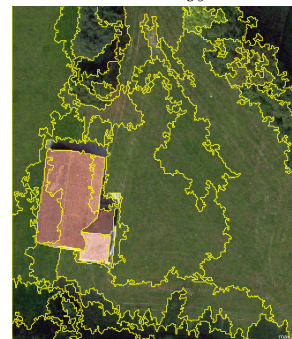
(f) Ergebnis Phase 2-4 mit *Local Removal Strategy*.



(g) Ergebnis Phase 6 mit *Universal Removal Strategy*.



(h) Ergebnis Phase 6 mit *Global Removal Strategy*.



(i) Ergebnis Phase 6 mit *Local Removal Strategy*.

**Abb. 2.16.:** Anwendungsbeispiele der Verfahren zur Bereinigung inkonsistenter Segmente.

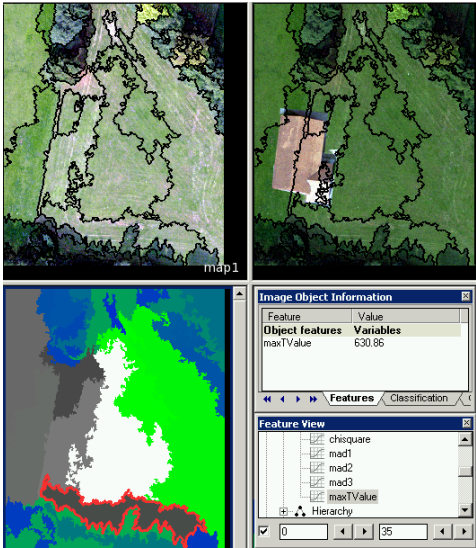


Abb. 2.17.: Wahl des Parameters  $T_i$  anhand der Bilddaten.

Segmentation of first image	
Map	map1
Level Name	New Level
Layerweights 1	1, 1, 1, 0, 0, 0, 0
Scale parameter	50
Shape Weight	0.1
Compacness Weight	0.5
Mutual Tolerance	0
Consistency Test	
Map	mapCT
Level Name	New Level
Consistency Test	Threshold Test
Removal Strategy	Universal Removal Strategy
Minimum Tree Depth	2
Layerweights test	0, 0, 0, 1, 1, 1, 0, 0
Scale parameter	50
maximum TVValue	
Scale parameter (relative)	8
Shape Weight	0.1
Compacness Weight	0.5
Mutual Tolerance	0

Segmentation of second image masking the changed regions	
Map	map2mask
Level Name 2 (masked)	New Level
Layerweights 2 (masked)	0, 0, 0, 1, 1, 1, 0, 0
Scale parameter	50
Shape Weight	0.1
Compacness Weight	0.5
Mutual Tolerance	0
Segmentation of second image	
Map	map2
Level Name 2	New Level
Layerweights 2	0, 0, 0, 1, 1, 1, 0, 0
Scale parameter	15
Shape Weight	0.1
Compacness Weight	0.5
Mutual Tolerance	0

Abb. 2.18.: Ausschnitt der Eingabemaske des MRS4CD Plugins für *eCognition*.

ist dies essentiell.

Die erste Parameterkategorie mit dem Titel **Segmentation of first Image** wurde bereits auf Seite 59 beschrieben. Sie beinhaltet lediglich die entsprechenden Parameter für den *Multiresolution-Segmentation*-Algorithmus für Einzelbilder **Scale Parameter** für  $T$ , **Shape Weight** für  $w_s$ , **Compactness Weight** für  $w_{comp}$  und **Mutual Tolerance**. Welche Bildkanäle dazu verwendet werden sollen, wird in Parameter **Layerweights** 1 definiert. Dieser Teil des Eingabeformulars deckt Phase 1 ab (siehe Tabelle 2.1). Im Beispiel in Abbildung 2.18 sind die Parameter so gewählt, dass für den ersten Aufnahmezeitpunkt die drei Kanäle des ersten Bildes mit gleicher Wichtung eingesetzt und mit einem Schwellenwert von  $T = 50$  segmentiert werden. Die Gewichtsparameter  $w_s = 0.1$  und  $w_{comp} = 0.5$  sind so gewählt, dass die Segmentierung nur zu 10% auf Formeigenschaften beruht und dieser Anteil wiederum gleichmäßig auf die Kompaktheit und Glattheit der Segmente verteilt wird. Die Ergebnisse der Segmentierung werden in eine *map* mit dem Namen **map1** in der Hierarchieebene **New Level** abgelegt.

Der zweite Bereich des Eingabeformulars mit dem Titel **Consistency Test** umfasst die wesentlichen Parameter des in diesem Abschnitt vorgestellten Algorithmus. In Tabelle 2.1 entspricht dies den Phasen 2, 3 und 4. Zunächst werden die Parameter **Map** und **Level Name** gesetzt. Das Beispiel in Abbildung 2.18 verwendet hierzu die *map* mit dem Namen **mapCT** und die Segmentebene **New Level**. Danach wählt der Nutzer im Parameter **Consistency Test** eine der vier beschriebenen Methoden zur Überprüfung der Konsistenz, im Beispiel in Abbildung 2.18 den *Threshold Test*. Anschließend kann mithilfe des Parameters **Removal Strategy** eines der drei Verfahren zur Entfernung inkonsistenter Segmente gewählt werden. Im Beispiel in Abbildung 2.18 kommt hier die *Universal Removal Strategy* zum Einsatz. Mit **Minimum Tree Depth** definiert der Nutzer, ab welcher Tiefe des Segmentbaumes mit der Konsistenzüberprüfung begonnen werden soll. Wenn dieser Parameter wie im Beispiel in Abbildung 2.18 auf den Wert 2 gesetzt ist, so beginnt die Prüfung direkt oberhalb der Ebene einzelner Pixel. In **Layerweights test** wird festgelegt, welche Bildkanäle für die Konsistenzüberprüfung verwendet werden sollen. Dabei ist anzumerken, dass nicht nur die Bildkanäle des zweiten Bildes hier angegeben werden können, sondern dass es auch möglich ist, den Intensitätskanal einer pixel-basierten Änderungsdetektion als Grundlage zu verwenden. Im Beispiel in Abbildung 2.18 kommen dazu die drei Bildkanäle des zweiten Aufnahmezeitpunkts mit gleicher Wichtung zum Einsatz. Der Parameter **Scale parameter** entspricht dem Parameter  $T_t$  im Modell, worüber die Sensitivität der Methode zur Konsistenzüberprüfung eingestellt werden kann. Abbildung 2.18 verwendet hier denselben Wert wie für den Schwellenwert zur Segmentierung des ersten Aufnahmezeitpunkts. Optional ist es möglich, im Parameter **maximum Tvalue** eine Objektvariable zu definieren, in der gespeichert wird, welchen Wert die Testgröße, d.h. die linke Seite der Gleichung 2.57, hat, falls ein Segment nicht aufgeteilt wird. Über **Scale parameter (relative)** wird, falls der Wert **Relative Threshold Test** für den Parameter **Consistency Test** ausgewählt ist, die Größe  $T_{rt}$  bestimmt. **Shape Weight** und **Compactness Weight** liefern die Werte für  $w_s$  bzw.  $w_{comp}$ . In Abbildung 2.18 kommen hier dieselben Werte wie für die Segmentierung des ersten Aufnahmezeitpunkts zum Einsatz.

In der dritten und vierten Parameterkategorie **Segmentation of second image masking the changed regions** und **Segmentation of second image** werden analog zur ersten Parameterkategorie die Werte für den *Multiresolution-Segmentation*-Algorithmus der Phasen 5 und 6 eingegeben. Das Beispiel in Abbildung 2.18 verwendet hier die *maps* mit den Namen **map2mask** und **map2**. Die Schwellenwerte für den MRS-Algorithmus werden auf die Werte  $T = 50$  für die Phase 5 und  $T = 15$  für die Phase 6 gesetzt. Die Parameter für die Gewichte  $w_s$  und  $w_{comp}$  werden aus den vorherigen Phasen des Algorithmus unverändert übernommen.

In Abbildung 2.19 ist ein Beispiel für das Ergebnis des Algorithmus zu sehen. Dafür wurde der *Relative Threshold Test* in Kombination mit der *Universal Removal Strategy* eingesetzt. Beide Bilder unterscheiden sich bis auf synthetisch hinzugefügtes Rauschen nur am linken unteren Rand voneinander. Die Segmentationen des ersten und zweiten Aufnahmezeitpunktes weisen

Unterschiede nur in diesem Bereich auf. Gleichzeitig sind die Segmentationen sehr gut an die jeweiligen Aufnahmen angepasst.

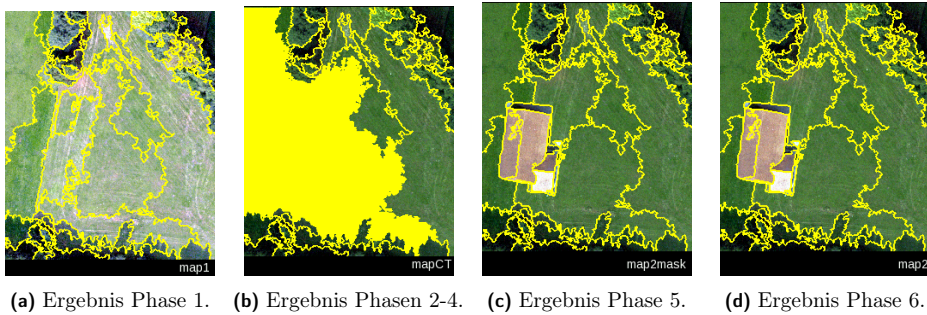


Abb. 2.19.: Beispielergebnis des MRS4CD Plugins für *eCognition*.

## Implementierung

Nach eingehender Betrachtung des Segmentierungsalgorithmus sowie der Anwendung des Plugins soll im Folgenden auf einige Details der Implementierung eingegangen werden. Im Abschnitt 2.3.3 wurde die Reimplementierung des *Multiresolution-Segmentation*-Algorithmus erläutert. Diese Reimplementierung bildet auch die Grundlage für die Implementierung des MRS4CD-Algorithmus. Insbesondere die Klassenstruktur, welche in 2.4 dargestellt wird, ist für diese Implementierung von großer Bedeutung.

Um den MRS4CD-Algorithmus zu implementieren, wurde das Modell in erster Linie in der Klasse **Segment** erweitert. Einerseits existiert für jede Methode zur Konsistenzüberprüfung in der Klasse **Segment** eine Objektmethode mit dem Präfix **checkConsistencyBy**, die prüft, ob ein Segment diesen Test unter Berücksichtigung der aktuell gewählten Daten erfüllt. Über die drei beschriebenen Methoden zur Konsistenzüberprüfung hinaus sind auch weitere Methoden im Klassenmodell enthalten, die jedoch nur testweise während der Entwicklung implementiert wurden und in der endgültigen Version des Plugins keine Anwendung finden. Andererseits gibt es für jede der drei Strategien zur Entfernung inkonsistenter Segmente ebenfalls jeweils eine Methode in der Klasse **Segment**, deren Namen mit **remove** beginnt.

Die implementierte Version des MRS4CD-Verfahrens beginnt für die Phase 1 (siehe Tabelle 2.1) mit der Ausführung der Methode **runMRS**, um das Bild des ersten Aufnahmezeitpunkts zu segmentieren. Danach wird die Methode **adaptToImage** in der Klasse **Segmentation** verwendet, um die Überprüfung der Konsistenz und Entfernung inkonsistenter Segmente, d.h. die Phasen 2 bis 4 des Verfahrens, abzudecken. Innerhalb **adaptToImage** werden zunächst mit **setParameters** die im Formularbereich **Consistency Test** übergebenen Parameter eingelesen sowie alle Farbeigenschaften der Segmente durch **recalcHeterogeneity** neu berechnet. Anschließend wird für jedes Segment mittels **isConsistent** in **Segmentation** die jeweilige Methode zur Konsistenzüberprüfung in den **checkConsistencyBy**-Methoden der Klasse **Segment** aufgerufen. Danach werden ggf. durch die Methode **cleanUp** und darin wiederum durch Aufruf der jeweiligen **remove**-Methode die inkonsistenten Segmente entfernt.

In der folgenden Phase 5 des MRS4CD-Algorithmus wird mit **runMRS** der *Multiresolution-Segmentation*-Algorithmus auf den maskierten Bereichen ausgeführt. Anschließend wird mit **setAllSegmentMasksToValue** diese Maskierung in Phase 6 aufgehoben. Mit einem weiteren Lauf der Methode **runMRS** auf dieser Grundlage endet die Segmentierung des Bildpaares. Für die Implementierung werden zwei Konzepte zur Datenhaltung benötigt, die im Folgenden



erläutert werden sollen.

Das erste Konzept realisiert die Speicherung des Segmentbaumes. Denn um das MRS4CD-Verfahren ausführen zu können, wird nicht nur das Segment der obersten Hierarchieebene benötigt, sondern auch jedes Zwischenergebnis. Dazu existieren in der Klasse **Segment** die Eigenschaften **parent**, **child1** und **child2**. Diese drei Attribute werden hauptsächlich bei der Entfernung inkonsistenter Segmente benötigt. Für die *Universal Removal Strategy* kann über das **parent**-Attribut zu einem inkonsistenten Segment das zugehörige Segment der obersten Hierarchieebene gefunden werden. Analog erhält der Algorithmus für die beiden anderen Bereinigungsverfahren die jeweils benötigten Elemente des Segmentbaumes.

Ein zweites wichtiges Konzept ist die Speicherung von Nachbarschaftsbeziehungen der untersten Hierarchieebene, den LLNHR-Objekten. Anstatt die Objekte vom Typ **NeighbourhoodRelation** auf jeder Ebene des Segmentbaumes auch nach der Fusion eines Segments komplett zu speichern, wird für jedes Segment, das sich nicht mehr in der obersten Hierarchieebene befindet nur eine Referenzliste mit Verweisen auf **NeighbourhoodRelation**-Objekte zwischen Pixelsegmenten gehalten. Dies ermöglicht die korrekte Etablierung von Nachbarschaftsbeziehungen nachdem Modifikationen an den Segmentbäumen vorgenommen wurden. Eine simple Speicherung der **NeighbourhoodRelation**-Objekte würde dazu nicht ausreichen, weil durch die Entfernung inkonsistenter Segmente in Phase 4 können auf einmal Segmente benachbart sein, die vorher noch nie benachbart waren und somit zu deren Nachbarschaft zuvor auch kein **NeighbourhoodRelation**-Objekt vorhanden war<sup>5</sup>. Umgesetzt wird das Konzept, indem bei jeder Fusion zweier Segmente  $s_1$  und  $s_2$  zum neuen Segment  $s_3$  die symmetrische Differenz der Datenstrukturen **lowlevelneighbours** von  $s_1$  und  $s_2$  gebildet und diese in **lowlevelneighbours** von  $s_3$  abgelegt wird. Definiert man  $LLN(s_i)$  als die Menge der LLNHR-Objekte, die in **lowlevelneighbours** gespeichert werden, so gilt formal, dass

$$\begin{aligned} LLN(s_3) &= LLN(s_1) \setminus \overline{LLN(s_2)} \cup LLN(s_2) \setminus \overline{LLN(s_1)} \\ &= LLN(s_1) \triangle LLN(s_2). \end{aligned} \tag{2.59}$$

Die beschriebene Herangehensweise ist zielführend, da bei der Bildung der symmetrischen Differenz nur jene **NeighbourhoodRelation**-Objekte aufgenommen werden, die Nachbarschaftsbeziehungen außerhalb von  $s_3$  abbilden.

Die Verwendung des Konzepts bei der Entfernung eines Segments aus dem Segmentbaum wird in Abbildung 2.20 beispielhaft anhand des Aufteilens von Segment  $T_2$  in seine Kindsegmente  $I_1$  und  $I_2$  dargestellt. Die während der Fusion entfernten Nachbarschaftsbeziehungen werden dabei auf Grundlage der LLNHR-Objekte rekonstruiert, d.h. zwischen  $I_1$  und  $T_1$  auf Basis von  $L_3-L_1$ , zwischen  $I_2$  und  $T_3$  auf Basis von  $L_2-L_4$  und zwischen  $I_1$  und  $I_2$  auf Basis von  $L_1-L_2$ . Die Abkürzungen  $L_i$  stehen dabei für Segmente der untersten Hierarchieebene sogenannte *Low-level Segments* (LLS).  $I_x$ . Die Segmente  $I_j$  repräsentieren intermediäre, d.h. während des Algorithmus zwischenzeitlich vorhandene, Segmente. Die Segmente  $T_k$  schließlich verkörpern Segmente der obersten Hierarchieebene, die *Top-level Segments*.

## Komplexität und Ressourcenbedarf

Die Intention der präsentierten Implementierung liegt in erster Linie darin zu zeigen, dass das MRS4CD-Verfahren umgesetzt werden kann. Weniger Wert wurde auf eine Optimierung der Laufzeit und des Speicherbedarfs für den operationellen Einsatz des Verfahrens gelegt. Trotzdem ist es wichtig, die Auswirkungen der Parameter des Verfahrens auf die Laufzeit und den

<sup>5</sup>Prinzipiell ist es sogar möglich, auf die Speicherung der LLNHR-Listen zu verzichten und diese ad hoc zu ermitteln, wenn sie benötigt werden. Während der Implementierung des Verfahrens hat sich allerdings herausgestellt, dass dieser Ansatz zu viel Rechenzeit benötigt.

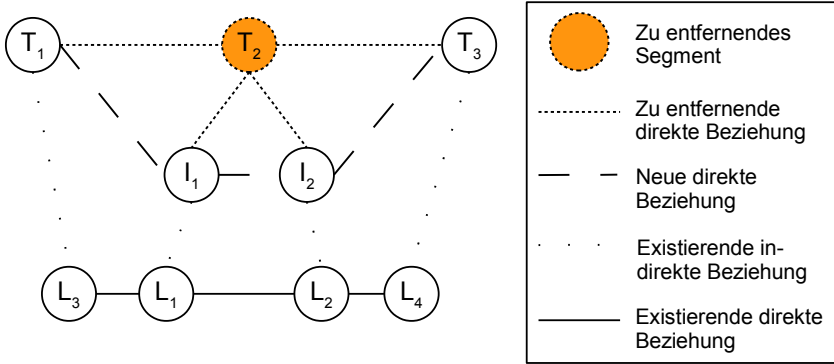


Abb. 2.20.: Konzept zur Speicherung der Nachbarschaftsbeziehungen transientser Segmente.

Speicherbedarf zu untersuchen, um stark ansteigenden Ressourcenbedarf erklären und ggf. beheben zu können. Daher soll im Folgenden zunächst eine theoretische Komplexitätsuntersuchung des Algorithmus erfolgen. Anschließend wird die Laufzeit anhand eines praktischen Beispiels analysiert.

Um eine Analyse der Komplexität des Algorithmus durchführen zu können, wird die Definition von sechs Parametern benötigt. Vor Durchführung der Phase 1 (siehe Tabelle 2.1) seien  $n$  Pixel in  $k$  Bildkanälen gegeben, woraus der MRS-Algorithmus  $m$  TLS-Objekte erzeugt. Die Phasen 2 bis 4 teilen diese  $m$  TLS-Objekte in  $n'$  TLS-Objekte auf unter Verwendung von  $k'$  Bildkanälen. Abschließend verbinden die Phasen 5 und 6 des Algorithmus die  $n'$  TLS-Objekte zu  $m'$  TLS-Objekten der endgültigen Segmentation unter Verwendung von  $k''$  Bildkanälen. Die Komplexitätsanalyse kann vereinfacht werden, indem die maximale Anzahl der in den einzelnen Phasen des Algorithmus verwendeten Kanäle als

$$k_{max} = \max\{k, k', k''\} \quad (2.60)$$

sowie die minimale Anzahl der Segmente der obersten Hierarchieebene am Ende der Phasen 1 und 6 als

$$m_{min} = \min\{m, m'\} \quad (2.61)$$

definiert werden.

Für die Betrachtung der Laufzeitkomplexität des MRS4CD-Verfahrens ist es nötig, die Phasen 1 bis 6 des Algorithmus näher zu untersuchen. Dabei stellt sich heraus, dass die Laufzeit der Phasen 1, 5 und 6 mit der Laufzeit einer zweifachen Ausführung des MRS-Algorithmus abgeschätzt werden, d.h. es gilt

$$C_{\text{MRS4CD } P_1}^{(r)} = \mathcal{O}(k(n^3 - m^2(m-1))) \quad (2.62)$$

und

$$C_{\text{MRS4CD } P_5}^{(r)} + C_{\text{MRS4CD } P_6}^{(r)} = \mathcal{O}(k''(n'^3 - m'^2(m'-1))). \quad (2.63)$$

In der Phase 2 wird für alle **Segment**-Objekte die Heterogenität neu berechnet. Die Formeigenschaften müssen nicht berücksichtigt werden, da bisher keine Formänderung der Segmente stattgefunden hat. Zu diesem Zeitpunkt sind  $2n - m$  **Segment**-Objekte vorhanden. Für diese Objekte müssen die Farbeigenschaften der  $k$  Bildkanäle der Phase 1 gelöscht und anschließend für die  $k'$  Bildkanäle der Phase 2 neu berechnet werden. Dadurch entsteht ein Aufwand von

$$C_{\text{MRS4CD } P_2}^{(r)} = \mathcal{O}((2n - m)(k + k')) = \mathcal{O}(k_{max}(2n - m)) \quad (2.64)$$

für die Phase 2.

In der Phase 3 werden alle  $n - m$  Segmente (ohne Pixelsegmente) auf Konsistenz geprüft. Der *Threshold Test* und der *Relative Threshold Test* erzeugen dabei einen konstanten Rechenaufwand je Segment. Der *Local Best Fitting Test* sowie der *Local Mutual Best Fitting Test* benötigen hingegen pro zu überprüfendem Segment einen Aufwand von  $\mathcal{O}(kn)$ , was der Komplexität eines einzelnen bzw. doppelten Aufrufs der Methode `findNearestNeighbour` entspricht (siehe 2.31). Somit gilt für die Laufzeitkomplexität der Phase 3, dass

$$C_{\text{MRS4CD } P_3}^{(r)} = \mathcal{O}(kn(n - m)). \quad (2.65)$$

Bei der Bereinigung inkonsistenter Segmente in der sich anschließenden Phase 4 werden unabhängig von der verwendeten Methode im wesentlichen  $n' - m$  Segmente gelöscht und für jedes gelöschte Segment höchstens  $2n' + 1$  Nachbarschaftsobjekte etabliert. Das bedeutet für die Komplexität der Phase 4 einen Aufwand von

$$C_{\text{MRS4CD } P_4}^{(r)} = \mathcal{O}(n'(n' - m)). \quad (2.66)$$

Insgesamt folgt somit eine Laufzeitkomplexität für den MRS4CD-Algorithmus von

$$\begin{aligned} C_{\text{MRS4CD}}^{(r)} &= \sum_{i=1}^6 C_{\text{MRS4CD } P_i}^{(r)} \\ &= \mathcal{O}(k_{\max}(n^3 - m_{\min}(m_{\min} - 1)) + k_{\max}(2n - m) + kn(n - m) + n'(n' - m)) \\ &= \mathcal{O}(k_{\max}(n^3 - m_{\min}(m_{\min} - 1))). \end{aligned} \quad (2.67)$$

Die in Gleichung 2.35 ermittelte Abschätzung für die Laufzeitkomplexität des MRS-Algorithmus kann somit auch für den MRS4CD-Algorithmus verwendet werden.

Die Komplexität des Speicherbedarfs für den MRS4CD-Algorithmus muss im Gegensatz zur Laufzeit nicht näher betrachtet werden, weil nur eine Reduktion und ggf. eine Erweiterung der Segmentation auf Grundlage des MRS-Algorithmus erfolgt. Die Überprüfungs- und Bereinigungsschritte des Algorithmus erhöhen dabei temporär den Speicherbedarf nicht stärker, als während des MRS-Algorithmus selbst. Eine Ausnahme besteht jedoch im Einfluss der Kanalanzahl, weshalb erneut die Größe  $k_{\max}$  aus 2.60 zum Einsatz kommt. Somit gilt

$$C_{\text{MRS4CD}}^{(m)} = \mathcal{O}(n(n + k_{\max})). \quad (2.68)$$

In der Implementierung des Algorithmus kann jedoch sowohl der reale Speicherbedarf als auch die gemessene Laufzeit insbesondere durch die Fragmentierung des Hauptspeichers größer als beim MRS-Algorithmus sein. Beispielhaft wird dieser Effekt in Tabelle 2.2 gezeigt. Darin wurde der bereits in Abschnitt 2.3.3 verwendete Datensatz genutzt. Für das Bild des zweiten Aufnahmezeitpunktes wurde ein Haus aus einem Bildbereich in einen anderen kopiert sowie Gaußsches Rauschen mit  $\sigma = 5$  hinzugefügt.

Fallbeschreibung	Phasen	Laufzeit	Speicherbedarf
Nur erster Aufnahmezeitpunkt	1	36 s	783 MB
Beide Aufnahmezeitpunkte ohne Aufteilung	1,2,3	40 s	803 MB
Beide Aufnahmezeitpunkte mit vollständiger Aufteilung ohne abschließende Fusionen	1,2,3,4	11 min	953 MB
Zwei Bilder mit vollständiger Aufteilung und abschließende Fusionen	1,2,3,4,5,6	15 min	1268 MB

**Tab. 2.2.:** Exemplarische Übersicht der Laufzeit des Algorithmus und dessen Speicherbedarf.

### 2.3.5. Segmentierung großer Bilder

Die Abschnitte 2.3.4 und 2.3.3 haben den erhöhten Bedarf an Rechenleistung und Speicher des MRS- und MRS4CD-Algorithmus gezeigt. Insbesondere die Größe des Bildes spielt dabei eine entscheidende Rolle.

Der Bedarf an Rechenleistung erweist sich als weniger gravierend als der erhöhte Speicherbedarf. Als diese Arbeit begonnen wurde, war *eCognition* nur als 32-Bit-Version verfügbar. Daraus resultierte eine Begrenzung des Hauptspeichers auf 2 GB pro Prozess. Die Kantenlänge eines unkomprimierten quadratischen Bildes mit vier Kanälen und 16 Bit pro Kanal durfte somit 16384 Pixel nicht übersteigen. Dem könnte man entgegenhalten, dass die Bilddaten nicht permanent vollständig im Speicher gehalten werden müssen. Jedoch verlangt die Struktur des MRS-Algorithmus anfangs eine Segmentierung, bei der jedem Pixel ein Objekt entspricht. In diesen Objekten wiederum müssen die Farbinformationen gespeichert werden. De facto entspricht dies einem Ablegen des gesamten Bildes im Hauptspeicher.

Dem beschriebenen Problem kann begegnet werden, indem das Bild in kleinere Teile, sogenannte *Tiles* oder Kacheln, aufgeteilt wird. Die Kacheln werden unabhängig voneinander verarbeitet und anschließend wieder zum Gesamtbild zusammengefügt. Man nennt diese Herangehensweise *Tiling-Stitching* (TS).

Eine simple Methode der TS-Segmentierung kann in vier Schritten erfolgen:

1. Teile das Gesamtbild in Kacheln geringer Größe auf, so dass die einzelnen Kacheln problemlos im Speicher gehalten werden können.
2. Segmentiere die Kacheln einzeln mit den gegebenen Parametern.
3. Füge die Segmentation der Kacheln wieder zusammen.
4. Segmentiere von dieser Segmentation ausgehend ein weiteres Mal, um die endgültige Segmentation des Bildes zu erhalten.

Unerfreulicherweise kommt es dabei vor, dass die Bereiche, an denen Kacheln aneinander angrenzen, nicht korrekt segmentiert werden, wie in Abbildung 2.21 deutlich zu sehen ist.



**Abb. 2.21.:** Problem der Kachelgrenzen bei der direkten Anwendung der *Tiling-Stitching*-Methode.

Eine Lösung dieses Problems ist laut O'Neil-Dunne u. a. (2012) die Verwendung von stark überlappenden Kacheln. Dieser Ansatz führt jedoch zu erheblichem rechnerischem Mehraufwand, da eine Überlappung von ca. 60% empfohlen wird. Zudem konnte im Rahmen dieser Arbeit nicht festgestellt werden, wie die überlappenden Kacheln wieder konsistent zusammengefügt werden können.

Ein etwas komplizierteres Segmentierungsverfahren, das den TS-Ansatz verwendet, findet sich in Happ u. a. (2010). Darin wird eine parallelisierte Version des MRS-Algorithmus präsentiert. Das Ausgangsbild wird dabei in den von allen Threads gemeinsamen genutzten Speicher geladen. Anschließend werden Kacheln ohne Überlappung erzeugt und jedem Thread eine Kachel zugeordnet. Der Prozess iteriert in mehreren Durchläufen solange Kandidaten zur Fusion gefunden werden. (In einem Durchlauf wird jedes Segment einer Kachel außer den Segmenten an den Grenzen der Kachel durch den zugehörigen Thread genau einmal betrachtet und evtl. fusioniert. Durch Fusionen entstandene Objekte werden erst im nächsten Durchlauf erneut betrachtet.) Die Besonderheit des Verfahrens ist die Behandlung der Segmente an den Grenzen der Kachel. Obwohl diese zumindest anfangs einer Kachel zugeordnet sind, werden sie erst am Ende eines Durchlaufs durch einen einzelnen Thread für das gesamte Bild betrachtet. Der Vorteil des Verfahrens ist der Geschwindigkeitsgewinn durch die teilweise Parallelisierung des Algorithmus. Auf der anderen Seite zeigen sich jedoch auch zwei Nachteile. Erstens wird das Bild komplett in den Speicher geladen, was zur bereits angesprochenen Speicherproblematik führt. Zweitens muss in jedem Durchlauf auf die Segmentierung der Kachel gewartet werden, die am längsten benötigt, damit abschließend die Segmente an den Kachelgrenzen betrachtet werden können.

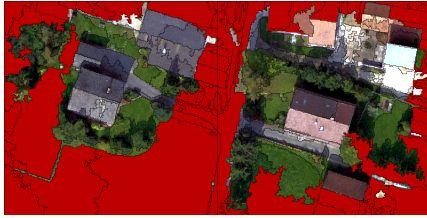
Da beide in diesem Abschnitt bisher vorgestellten Ansätze keine zufriedenstellenden Ergebnisse liefern, wird im Folgenden ein Verfahren zur *Tiling-Stitching*-Segmentierung mit multiplen Layern unter Einsatz des MRS-Algorithmus präsentiert.

Die Methode teilt zunächst das Bild in einem ersten Schritt ohne Überlappung in Kacheln auf (*Tiling*).

Im zweiten Schritt, der Kachel-Segmentierung, werden die Kacheln unabhängig voneinander mit denselben Parametern segmentiert. Wichtig ist dabei die Erzeugung mehrerer Hierarchieebenen für die Behandlung der Effekte an den Kachelgrenzen. Zusätzlich ist die Klassifizierung der Segmente mit einer Außenkante an einer der Kachelgrenzen als „Grenzsegmente“ nötig. Das Ergebnis dieses Schrittes wird in Abbildung 2.22 für zwei der fünf Hierarchieebenen dargestellt. Die als „Grenzsegmente“ klassifizierten Segmente werden darin rot umrandet dargestellt.

Die sich anschließende dritte Phase, die Aufteilung, ersetzt ausgehend von der obersten Hierarchieebene alle markierten Segmente durch die sie konstituierenden Segmente der darunterliegenden Hierarchieebene. Anschließend wird die darunterliegende Hierarchieebene gelöscht. Segmente in der resultierenden Segmentation, deren Elternobjekt der Klasse „Grenzsegmente“ angehörten, die aber selbst nicht an der Kachelgrenze liegen, werden als „ehemalige Grenzsegmente“ klassifiziert. Dieser Vorgang wiederholt sich, bis nur noch eine Hierarchieebene pro Kachel übrigbleibt. In Abbildung 2.23 wird der Ablauf des dritten Verfahrensschrittes beispielhaft dargestellt. Objekte der Klasse „Grenzsegmente“ werden darin rot, Objekte der Klasse „ehemalige Grenzsegmente“ gelb umrandet dargestellt.

Im vierten Teil des Algorithmus, dem *Stitching*, werden nun die Kacheln mitsamt ihren Segmentationen zusammengefügt. Alle Segmente der Klasse „ehemalige Grenzsegmente“ werden dabei wieder der Klasse „Grenzsegmente“ zugeordnet. Danach führt das Verfahren den MRS-Algorithmus mit dem *Scale*-Parameter der höchsten Hierarchieebene aus. Als Kandidatensegmente werden nur Segmente der Klasse „Grenzsegmente“ zugelassen. Anschließend wird der MRS-Algorithmus erneut auf Grundlage dieser Segmentation mit demselben *Scale*-Parameter wie zuvor jedoch ohne Beschränkung der Kandidatenmenge aufgerufen. Ein Beispiel für das Ergebnis des vierten Teils des Verfahrens wird in Abbildung 2.24 im Vergleich zu einer sequenziell mit denselben Parametern erzeugten Segmentation gezeigt. Obwohl diese beiden Ergebnisse nicht in jedem Detail übereinstimmen, kann die Ergebnisqualität des parallelen Verfahrens durchaus mit jener sequenziellen Algorithmus konkurrieren.



(a) Ebene 5 erstellt mit  $T = 30$ .



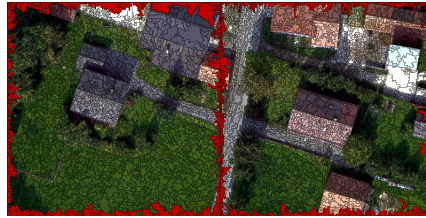
(b) Ebene 4 erstellt mit  $T = 25$ .



(c) Ebene 3 erstellt mit  $T = 15$ .

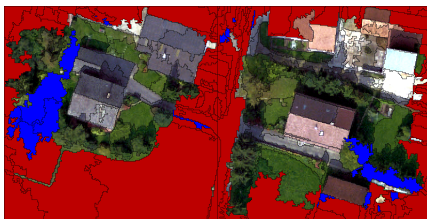


(d) Ebene 2 erstellt mit  $T = 10$ .

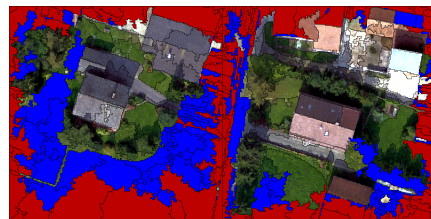


(e) Ebene 1 erstellt mit  $T = 5$ .

**Abb. 2.22.:** Fünf Hierarchieebenen einer Segmentation zweier Kacheln.



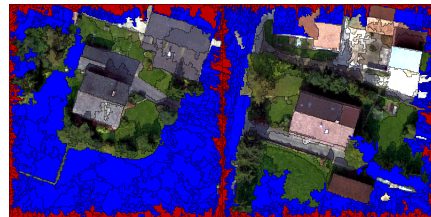
(a) 2.22a reduziert auf 2.22b.



(b) 2.23a reduziert auf 2.22c.



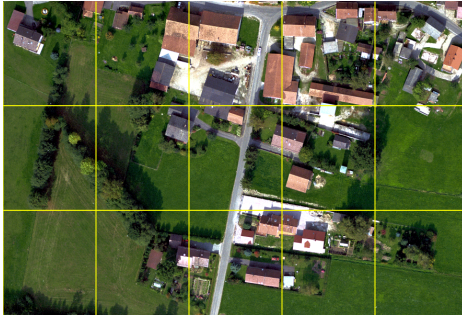
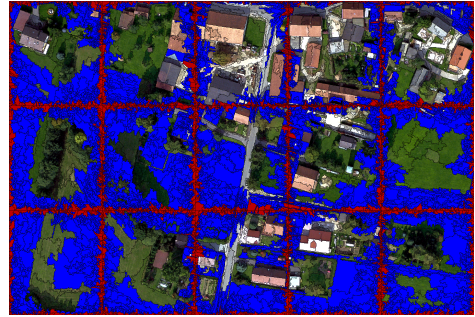
(c) 2.23b reduziert auf 2.22d.



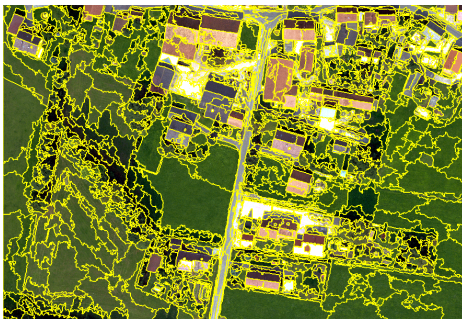
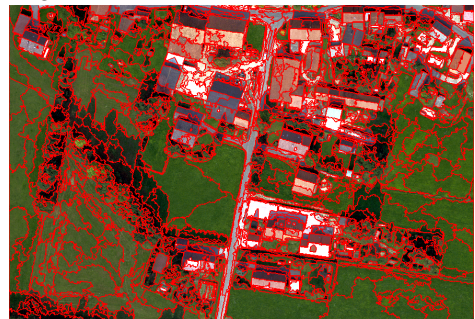
(d) 2.23c reduziert auf 2.22e.

**Abb. 2.23.:** Iterative Aufteilung der Grenzsegmente einer Segmentation auf die darunterliegenden Ebenen.

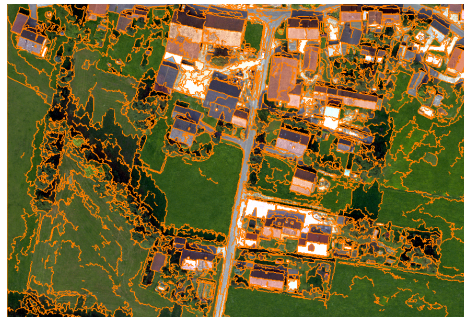


(a) Ergebnis *Tiling*.

(b) Ergebnis Kachel-Segmentierung und Aufteilung.

(c) Ergebnis *Stitching*.

(d) Ergebnis des sequenziellen MRS-Algorithmus.



(e) Gemeinsame Segmentgrenzen der Ergebnisse des TS-Verfahrens und des sequenziellen MRS-Algorithmus.

**Abb. 2.24.:** Überblick des TS-Verfahrens und Vergleich mit den Ergebnissen der sequenziellen Methode.



### 2.3.6. Zusammenfassung

In diesem Abschnitt wurde ein Segmentierungsalgorithmus vorgestellt, der für die objekt-basierte Änderungsdetektion eingesetzt werden kann.

Der Ansatz hat dabei drei wesentliche Vorteile. Erstens können nun die Formeigenschaften der Objekte in die Analyse einbezogen werden, da durch die Verwendung von unterschiedlichen Segmentationen für die diversen Aufnahmezeitpunkte Formänderungen zwischen den Aufnahmen zugelassen werden. Gleichzeitig besteht der zweite Vorteil des Verfahrens darin, dass diese Änderungen nur in jenen Bildbereichen auftreten, in denen auch wesentliche Unterschiede zwischen den Aufnahmen bestehen. Das Verfahren kann somit als stabil bezeichnet werden. Drittens wird durch den beschriebenen Ansatz eine hohe Anpassung der Segmentierung an die Daten erreicht, weil nicht mehr eine einzige Segmentierung beide Aufnahmen repräsentieren muss.

Für die Verwendung des Verfahrens ist es jedoch wesentlich, dass der Algorithmus in den Gesamtprozess der objekt-basierten Änderungsdetektion integriert werden kann. Eine Ansatz zur Lösung dieses Problems wird in Abschnitt 2.4.2 präsentiert.

## 2.4. Merkmalsextraktion

### 2.4.1. Einleitung

Nach Abschluss der Segmentierung folgt im Ablauf der objekt-basierten Änderungsdetektion die Merkmalsextraktion als nächster Schritt. Das Ziel dieser Phase ist einerseits die Verknüpfung der zwei Segmentationen, so dass im weiteren Verlauf sichergestellt ist, welche Segmente für den Vergleich der Bilder miteinander in Beziehung zu setzen sind. Dieses Thema ist Bestandteil des Abschnittes 2.4.2. Andererseits muss eine Auswahl der Objekteigenschaften getroffen werden, die für den Vergleich der Bilder herangezogen werden sollen. In Abschnitt 2.4.3 wird dieser Aspekt besprochen.

### 2.4.2. Objektentsprechung

Das in Abschnitt 2.3.4 vorgestellte Segmentierungsverfahren ist in wesentlichen Punkten den bisherigen Algorithmen zur Segmentierung im Rahmen der objekt-basierten Änderungsdetektion überlegen. Nichtsdestoweniger erzeugt das Verfahren zwei Segmentationen, die sich in einigen Bereichen voneinander unterscheiden können. Um jedoch eine Änderungsdetektion mit Methoden der multivariaten Statistik durchzuführen, wie sie in Abschnitt 2.5 vorgestellt werden, sind Zuordnungen zwischen den Objekten der Segmentationen notwendig. Im Detail liegt dies in der Tatsache begründet, dass Methoden der multivariaten Statistik die Objekte in den Bildern  $I_1$  und  $I_2$  mit den Segmentationen  $S_1$  und  $S_2$  als Realisierung von Zufallsvektoren  $\mathbf{X}$  bzw.  $\mathbf{Y}$  modellieren. Die Einträge der Vektoren ergeben sich durch die Merkmalsausprägungen. Die Länge der Vektoren resultiert entsprechend aus der Anzahl der verwendeten Merkmale  $p$ . Da die Merkmale dieser beiden Zufallsvektoren verglichen werden sollen, müssen zur Schätzung ihrer Verteilung Paare von Beobachtungen verwendet werden. Dazu werden im Folgenden zwei Methoden vorgestellt,

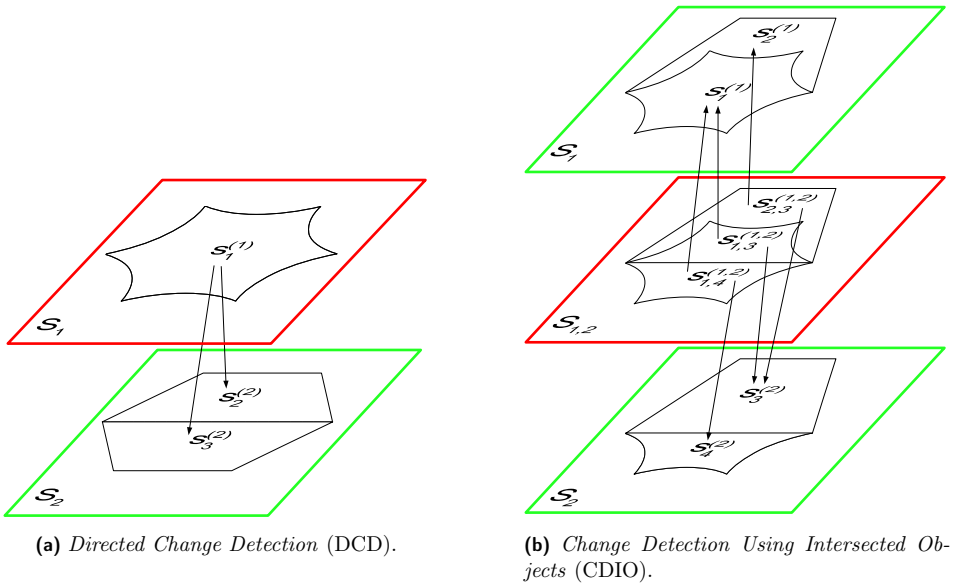
- die *Directed Change Detection* (DCD) und
- die *Change Detection Using Intersected Objects* (CDIO).

Beim ersten Ansatz, der DCD, werden jedem Segment  $s_i^{(1)}$  aus der Segmentierung  $S_1$  alle mit  $s_i^{(1)}$  räumlich überlappenden Segmente  $s_1^{(2)}, \dots, s_k^{(2)}$  aus  $S_2$  zugeordnet. Offensichtlich ergäbe

dies kein eindeutig bestimmtes Objektpaar, weshalb die Realisierungen der Zufallsvektoren  $\mathbf{X}$  und  $\mathbf{Y}$  bestimmt werden durch

$$\begin{aligned}\mathbf{x}_i &= (x_{i,1}, \dots, x_{i,p})^T = \mathbf{f}_{\mathbf{X}}(s_i^{(1)}), & i &= 1, \dots, n_1; \\ \mathbf{y}_i &= (y_{i,1}, \dots, y_{i,p})^T = \frac{1}{k} \sum_{j=1}^k \mathbf{f}_{\mathbf{Y}}(s_j^{(2)}), & i &= 1, \dots, n_1.\end{aligned}\tag{2.69}$$

In der Gleichung 2.69 stehen  $\mathbf{f}_{\mathbf{X}}$  und  $\mathbf{f}_{\mathbf{Y}}$  für Funktionen, deren Funktionswerte die Realisierungen der Merkmalsvektoren  $\mathbf{X}$  und  $\mathbf{Y}$  repräsentieren. Dadurch erhält man für jedes der  $n_1$  Segmente  $s_i^{(1)}$  aus  $S_1$  ein Tupel  $(\mathbf{x}_i, \mathbf{y}_i)$ , welches zum Schätzen der Verteilungsparameter von  $\mathbf{X}$  und  $\mathbf{Y}$  verwendet werden kann. In Abbildung 2.25a wird diese Methode der Objektentsprechung schematisch veranschaulicht.



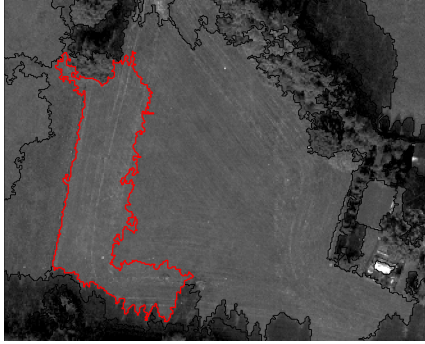
**Abb. 2.25.:** Zwei Methoden zur Lösung des Problems der Objektentsprechung.

Der Anwender muss sich bei dieser Methode bewusst sein, dass mitunter ein Segment  $s_i^{(1)}$  mit einem Segment  $s_j^{(2)}$  verknüpft wird, obwohl die Menge der gemeinsamen Pixel sehr gering ist. Die Forderung eines Mindestmaßes an Überlappung etwa in Prozent der Fläche von  $s_i^{(1)}$  oder auch in einer absoluten Pixelanzahl erscheint daher sinnvoll. Das kann jedoch bei ungünstigen Konstellationen zur Konsequenz haben, dass einem Segment  $s_i^{(1)}$  nicht ein einziges Segment  $s_j^{(2)}$  zugeordnet werden kann. Als Alternative kann daher die Benutzung eines mit der relativen Überlappung gewichteten Mittelwerts in Gleichung 2.69 vorgeschlagen werden.

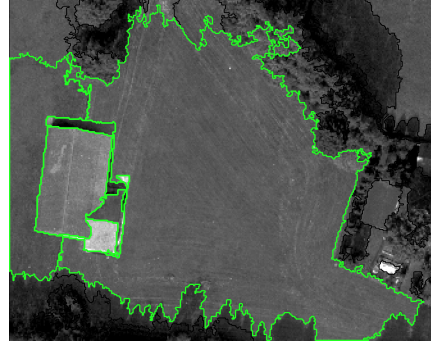
In *eCognition* kann DCD über den Prozess **create links** realisiert werden. Dabei werden Segmente einer **map** mit Segmenten einer zweiten **map** assoziiert. Als Parameter kann dem Prozess ein Schwellenwert für die mindestens geforderte Überlappung übergeben werden. Die Gleichung 2.69 realisiert man durch die Verwendung von Eigenschaften der Kategorie **Linked Object Statistics**<sup>6</sup>. Um die Implementierung des Prozesses der DCD besser zu veranschauli-

<sup>6</sup>Die Informationen der *eCognition*-Regelsätze werden in einem XML-Format gespeichert. Eigenschaften der Kategorie **Linked Object Statistics** können somit für einen einzelnen Kanal in der Programmumgebung

lichen, ist in Abbildung 2.26 ein Beispiel in *eCognition* gegeben.



(a) Segment in  $S_1$ , das mit dem Segment in  $S_2$  assoziiert ist.



(b) Segmente in  $S_2$ , die von dem Segment in  $S_1$  referenziert werden.

**Abb. 2.26.:** Beispiel für die Methode DCD.

Die zweite Methode zur Extraktion von korrespondierenden Objekten, die CDIO, ist motiviert durch die Arbeit zur klassifikations-basierte Änderungsdetektion in Stow (2010, S. 577), bei der jene Bereiche als Änderungsobjekte extrahiert werden, die unterschiedlichen Klassen zugeordnet sind.

Konkret ist die Idee hinter CDIO die Erzeugung einer dritten Segmentation  $S_{1,2}$ . Deren Segmente ergeben sich dann aus den räumlich zusammenhängenden Teilen des Durchschnitts der Segmente  $S_1$  und  $S_2$ . Formal bedeutet das

$$S_{1,2} = \left\{ s_{i,j,k}^{(1,2)} \mid i = 1, \dots, |S_1|; j = 1, \dots, |S_2|; k = 1, \dots, m_{i,j}; s_i^{(1)} \in S_1; s_j^{(2)} \in S_2 \right. \\ \left. s_{i,j,k}^{(1,2)} \text{ ist eine Zusammenhangskomponente von } s_i^{(1)} \cap s_j^{(2)} \right\}. \quad (2.70)$$

Eine Zusammenhangskomponente ist eine maximal große, nicht leere Teilmenge, die räumlich zusammenhängend ist (siehe auch Abschnitt 2.3.1). Mit  $m_{i,j}$  wird die Anzahl der Zusammenhangskomponenten angegeben, die bei der Bildung des Durchschnitts von  $s_i^{(1)}$  und  $s_j^{(2)}$  entstehen.

Da jedes Pixel des Bildbereichs in je genau einem Segment der Segmentationen  $S_1$  und  $S_2$  enthalten ist, gehört es auch genau einem Segment von  $S_{1,2}$  an. Daraus resultiert die Eindeutigkeit und Vollständigkeit der Segmentation. Die weiteren Eigenschaften der Definition einer Segmentation in 2.1 werden durch die Forderung nach nichtleeren Segmenten sowie dem räumlichen Zusammenhang in 2.70 erfüllt. Dass die Segmente homogen und maximal gemäß der Definition des Heterogenitätskriteriums in 2.2 sind, muss an dieser Stelle nicht gefordert werden, weil  $S_{1,2}$  nur als Hilfskonstrukt zur Integration von  $S_1$  und  $S_2$  dienen soll.

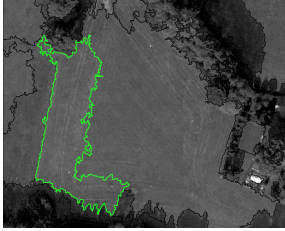
Schematisch ist die Funktionsweise in Abbildung 2.25b dargestellt. Ein Beispielergebnis des Verfahrens in *eCognition* findet sich in Abbildung 2.27. Realisiert wurde die Methode dabei, indem die Segmentationen  $S^{(1)}$  und  $S^{(2)}$  als Grauwertbilder mit den Segmentindizes als Pixelwert exportiert und wieder reimportiert wurden. Anschließend kann die Segmentationen  $S^{(1,2)}$  erzeugt werden, indem die Grauwertbilder als Eingabekanäle des MRS-Algorithmus genutzt werden und der Schwellenwert  $T = 0$  gesetzt wird. Für das Formgewicht muss dabei  $w_s = 0$  gelten.

Bereits zur Methode DCD wurde angemerkt, dass Segmentpaare  $s_i^{(1)} \in S_1$  und  $s_j^{(2)} \in S_2$ , die

---

definiert und anschließend mithilfe eines Texteditors für alle weiteren Kanäle dupliziert werden.

nur sich gegenseitig gering überlappen, das Ergebnis verfälschen können. Bei der Methode CDIO ist hierauf besonders zu achten, da durch kleine Fehler in der Koregistrierung der Bilder sogenannte Splitterpolygone entstehen. Laut Delafontaine u. a. (2009) eignen sich jedoch Formeigenschaften, um diese Segmente herauszufiltern.



(a) Segment  $s_i^{(1)}$  in  $S_1$ , das vom Segment  $s_{i,j}^{(1,2)}$  in  $S_{1,2}$  referenziert wird.



(b) Segment  $s_{i,j,1}^{(1,2)}$  in  $S_{1,2}$ , das mit den Segmenten  $s_i^{(1)}$  in  $S_1$  und  $s_j^{(2)}$  in  $S_2$  assoziiert ist.



(c) Segment  $s_j^{(2)}$  in  $S_2$ , das vom Segment  $s_{i,j}^{(1,2)}$  in  $S_{1,2}$  referenziert wird.

**Abb. 2.27.:** Beispiel für die Methode CDIO.

Die Realisierungen der Merkmalsvektoren  $\mathbf{X}$  und  $\mathbf{Y}$  können bei der Methode CDIO direkt ermittelt werden, da die Objektassoziationen eindeutig sind. Es gilt somit für ein beliebiges Segment  $s_{i,j,k}^{(1,2)} \in S^{(1,2)}$

$$\begin{aligned} \mathbf{x}_{i,j,k} &= \mathbf{f}_{\mathbf{X}}(s_{i,j,k}^{(1,2)}), \\ \mathbf{y}_{i,j,k} &= \mathbf{f}_{\mathbf{Y}}(s_{i,j,k}^{(1,2)}). \end{aligned} \tag{2.71}$$

Die beiden in diesem Abschnitt vorgestellten Methoden ermöglichen es, die objekt-basierten Änderungsdetektion auf der Grundlage von zwei unterschiedlichen Segmentationen durchzuführen. Neben der Bereitstellung eines stabilen Segmentierungsalgorithmus ist dies ein wesentlicher Bestandteil der Entwicklung eines integrierten Prozesses zur objekt-basierten Änderungsdetektion.

### 2.4.3. Merkmalsauswahl

Die Objekte, welche mit den bisher beschriebenen Verfahren extrahiert und miteinander assoziiert werden können, bedürfen für die weitere Verarbeitung im Prozess der Änderungsdetektion einer Beschreibung mithilfe von Eigenschaften. Im Gegensatz zu Pixeln, deren Eigenschaften bis auf Transformationen und Filterung die aufgenommenen Spektralwerte bilden, können Segmente mithilfe einer größeren Palette an Merkmalen beschrieben werden.

In Listner (2008) werden zur Kategorisierung der Merkmale fünf Rubriken vorgeschlagen:

1. **Spektrale Eigenschaften**, wie z.B. der Mittelwert oder andere deskriptive Statistiken eines Spektralkanals, sind eine direkte Erweiterung der Spektralwerte der einzelnen Pixel. Diese Eigenschaften ermöglichen die Betrachtung der Farbe eines Objekts sowie deren Verteilung innerhalb der Objektgrenzen.
2. **Textureigenschaften** können unter anderem aus der Grauwertübergangsmatrix abgeleitet werden (siehe Haralick u. a. 1973). Diese Merkmale versuchen, die Eigenschaften von sich wiederholenden Strukturen innerhalb eines Objekts zu quantifizieren. Ein großer Vorteil der Verwendung von Textureigenschaften gegenüber dem Einsatz von Texturfiltern in der pixel-basierten Bildanalyse liegt im Vorhandensein der Objektbegrenzung.

Dadurch werden Effekte an den Objektgrenzen vermieden und die Eigenschaftswerte auf diese Weise nicht verfälscht. Andererseits erscheint als Nachteil, dass die Größe der Grundgesamtheit zur Schätzung der GLCM-Statistiken wegen der variierenden Objektgrößen nicht konstant ist. Somit können zwei Objekte trotz gleicher Textur unterschiedliche Werte in den Texturmerkmalen haben, was einer unverfälschten Wiedergabe der Objekteigenschaften zuwiderläuft.

3. **Formeigenschaften** repräsentieren die Gestalt eines Objekts. Sie werden nicht mehr direkt aus den Beobachtungen des Sensors abgeleitet, sondern ergeben sich indirekt aus der Segmentierung. Ihr Wert hängt somit stark vom verwendeten Segmentierungsverfahren ab. Als Beispiele sind der Umfang, die Fläche oder die Rauheit der Begrenzungslinie zu nennen. Durch die in dieser Arbeit vorgestellte Methode zur Änderungsdetektion sind Formeigenschaften auch für die objekt-basierte Änderungsdetektion verwendbar.
4. **Hierarchie-basierte Eigenschaften** fußen auf der Idee, dass verschiedene Objekte auf verschiedenen Maßstäben sichtbar sind. Daher sollten für die Abbildung der einzelnen Maßstäbe unterschiedliche Segmentation herangezogen werden, die zueinander in einem hierarchischen Verhältnis stehen. Aus dieser Herangehensweise resultiert eine Baumstruktur der Segmente, die jener in Abschnitt 2.3.3 beschriebenen Struktur ähnelt. Beziehungen zwischen Segmenten der verschiedenen Hierarchieebenen führen dann zu abgeleiteten Eigenschaften wie zum Beispiel der Anzahl der Subsegmente und dem durchschnittlichen spektralen Unterschied zwischen einem Segment und seinen Subsegmenten. Im Rahmen dieser Arbeit spielt diese Art von Objekteigenschaften keine Rolle, da zur Vereinfachung für jeden Aufnahmezeitpunkt nur eine Segmentation herangezogen wird. Auf eine komplette Segmentationshierarchie und folglich auch auf hierarchie-basierte Eigenschaften wird im Rahmen dieser Arbeit verzichtet.
5. Mit **klassenbezogenen Eigenschaften** können semantische Zusammenhänge zwischen Objekten hergestellt werden. Wenn zum Beispiel ein als „Landmasse“ identifiziertes Objekt vollständig von Objekten der Klasse „Wasser“ umgeben ist, so kann abgeleitet werden, dass dieses Objekt der Klasse „Insel“ angehört. Für die vorliegende Arbeit ist auch diese Kategorie von Objekteigenschaften ohne Relevanz, da es sich bei der hier vorgestellten Methode zur Änderungsdetektion um einen Ansatz ohne Verwendung einer Klassifikation handelt.

Darüber hinaus lassen sich aus den elementaren Eigenschaftswerten mithilfe mathematischer Funktionen aggregierte Eigenschaften erstellen. Als Beispiel zu nennen sind die weit verbreiteten rationalen Funktionen wie etwa der *Normalized Differenced Vegetation Index* (NDVI) und dessen Varianten (Kriegler u. a. 1969).

Eine umfassende Darstellung der möglichen Objektmerkmale ist im Rahmen dieser Arbeit nicht vorgesehen. Für Details zu weiteren in *eCognition* verfügbaren Objekteigenschaften sei daher auf Definiens Imaging, Munich (2009), Niemeyer (2001, S. 145-146) und Marpu (2009, S. 60-66) verwiesen.

Wichtiger als die Auflistung der zur Analyse bereitstehenden Merkmale eines Objekts ist jedoch eine Anleitung, welche Objektmerkmale für eine Änderungsdetektion zu verwenden sind. Dazu kann die Überlegung angestellt werden, dass jene Eigenschaften verwendet werden sollten, die zu einem hohen Informationsgehalt der Merkmalsrealisierungen führen. Als Maß für den Informationsgehalt der Realisierungen eines Merkmals  $f$  in einer Segmentation  $S$  ist die Entropie  $E_S(f)$  geeignet (siehe z.B. Canty 2009, S. 62-65). Wenn  $h_S(v)$  die relative Häufigkeit eines Eigenschaftswertes  $v$  aus der Menge der möglichen Eigenschaftswerte  $V_f$  ist, so kann die Entropie definiert werden als

$$E(f) = - \sum_{v \in V_f} h_S(v) \ln h_S(v). \quad (2.72)$$

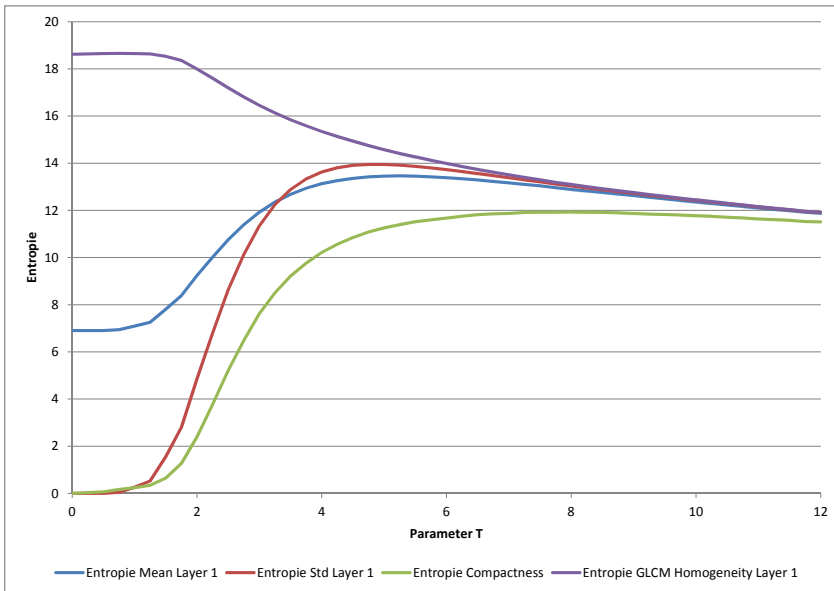
Für die Analyse sollten nun jene Merkmale  $f$  verwendet werden, deren Realisierungen einen hohen Informationsgehalt erwarten lassen, d.h. deren Entropie  $H(f)$  hoch ist.

In *eCognition* kann der Wert für die Entropie eines beliebigen Objektmerkmals nicht direkt berechnet werden. Daher wurde im Rahmen dieser Arbeit ein Plugin für *eCognition* implementiert, das die Berechnung durchführt. Die Eingabemaske des Plugins ist in Abbildung 2.28 dargestellt. Die Auswahl des zu verwendenden Objektmerkmals erfolgt über den Parameter **Feature**, die Szenenvariable wird in **Entropy variable** festgelegt. Außerdem ermöglicht das Plugin im Parameter **Output File** eine Datei zu bestimmen, in die der Entropie des gewählten Objektmerkmals für die Szene im CSV-Format exportiert wird. Auf der dieser Arbeit beiliegenden CD ist das Plugin im Verzeichnis **Entropy** abgelegt. Ein Beispiel zur Illustration der Verwendung des Plugins findet sich im Unterverzeichnis **example**.

Parameter	Value
Feature	Mean Layer 1
Entropy variable	entropyMeanLayer1
Output File	outputEntropy.csv

**Abb. 2.28.:** Eingabemaske des Algorithmus zur Entropiebestimmung eines Objektmerkmals.

Bei der Nutzung des Plugins ist zu beachten, dass der ermittelte Wert für die Entropie eines Merkmals auch von der Segmentation und dabei insbesondere vom Schwellenwert  $T$  abhängt. In Abbildung 2.29 ist dieser Effekt am Beispiel des in diesem Kapitel bereits mehrfach eingesetzten Datensatzes deutlich zu sehen. Da für  $T \rightarrow \infty$  die Objektanzahl gegen 1 geht, streben alle Entropiekurven gegen 0.



**Abb. 2.29.:** Entropie einiger Objektmerkmale in Abhängigkeit vom Parameter  $T$  des MRS-Algorithmus.

Weiterhin ist im Rahmen der Eigenschaftsauswahl für die weitere Analyse zu beachten, dass die Entropie der gewählten einzelnen Merkmale zwar hoch sein sollte. Falls die Merkmalsauswahl in sich jedoch stark korreliert ist, führt dies zu numerischen Problemen bei der Transformation. Die Verwendung einer geringeren Anzahl weniger stark korrelierter Merkmale ist in solchen Fällen angeraten. Mithilfe der Hauptkomponentenanalyse, die in Abschnitt 2.5.3 vor-

gestellt wird, ist es möglich, den Merkmalsraum so zu transformieren, dass die Eigenschaften untereinander unkorreliert sind. Auf diese Weise kann sichergestellt werden, dass die Merkmalsauswahl die Bildinformation bestmöglich abdeckt.

#### 2.4.4. Zusammenfassung

In diesem Abschnitt wurden einerseits zwei Konzepte vorgestellt und implementiert, mit denen die Segmentationen der beiden Aufnahmezeitpunkte für die weitere objekt-basierte Analyse integriert werden können. Darüber hinaus wurde gezeigt, wie die Werte von Objektmerkmalen bei dieser integrierten Segmentation ermittelt und den Segmenten zugeordnet werden können. Für eine weiterführende Diskussion der jeweiligen Vor- und Nachteile der präsentierten Methoden sei an dieser Stelle auf Abschnitt 3.2 verwiesen.

## 2.5. Transformation des Merkmalsraumes

### 2.5.1. Einleitung

Mit den Methoden aus Abschnitt 2.4 können die Daten der beiden Aufnahmezeitpunkte so aufbereitet werden, dass korrespondierende Objekte mit ihren Merkmalen als Elemente eines  $p$ -dimensionalen Vektorraums vorliegen. Diese lassen sich als Realisierungen von Zufallsvektoren in diesem Raum auffassen. Um anhand dieser Darstellung der Daten Hinweise auf Änderungen zu finden, sind strukturentdeckende Verfahren der multivariaten Statistik hilfreich. Im Rahmen der Fernerkundung werden solche Verfahren häufig eingesetzt. Zu nennen sind in diesem Zusammenhang beispielhaft die *Tasseled-Cap*-Transformation (Kauth und Thomas 1976), die Hauptkomponentenanalyse (Wiemker 1997), die Multivariate Änderungsdetektion (Nielsen u. a. 1998), die *Maximum Autocorrelation Factor Analysis* (MAF) (Nielsen u. a. 1998) sowie die Diskriminanzanalyse nach Fisher (Camps-Valls und Bruzzone 2005). Einen guten Überblick der mathematischen Grundlagen multivariater Statistik liefert Mardia u. a. (1979). Für eine anwendungsorientierte Perspektive in Hinblick auf die Fernerkundung sei jedoch auf Canty (2009) verwiesen.

Im Folgenden werden aus der Menge multivariater statistischer Analysemethoden die Hauptkomponentenanalyse sowie die Multivariate Änderungsdetektion näher betrachtet. Begründet ist diese Auswahl durch die hohe Qualität der Ergebnisse, die mit diesen Verfahren im Bereich der pixel- und objekt-basierten Änderungsdetektion bereits erzielt wurden (siehe z.B. Niemeyer 2001; Marpu 2009; Canty 2009; John 2009). Jedem der beiden Verfahren ist im Anschluss an eine kurze Einführung in die multivariate Datenanalyse jeweils ein Unterabschnitt gewidmet. Neben den theoretischen Grundlagen wird darin auf die Implementierung der Verfahren in *eCognition* eingegangen.

### 2.5.2. Multivariate Datenanalyse<sup>7</sup>

Mithilfe multivariater statistischer Methoden können Bildobjekte in Form von Vektoren dargestellt werden. Ein beliebiges Segment  $s_i$  kann somit durch den Vektor  $\mathbf{x}_i$  repräsentiert werden, der eine Ausprägung des Zufallsvektors  $\mathbf{X}$  ist. Die Komponenten des Vektors entsprechen dabei den Merkmalen, die auf diese Weise als Zufallsvariable modelliert werden. Die möglichen Werte der Merkmale sind wiederum Ausprägungen dieser Zufallsvariablen. Durch diese Darstellung kann man Bildobjekte wie multivariate Datensätze behandeln.

Es ist in der multivariaten Statistik üblich, alle gegebenen Objekte, d.h. die Stichprobe des

<sup>7</sup>Die Grundlage für diesen Abschnitt bildet Listner (2008).

Zufallsvektors  $\mathbf{X}$ , in einer Datenmatrix  $\mathbf{D}_\mathbf{x}$  als

$$\mathbf{D}_\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n) = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{p1} & \cdots & x_{pn} \end{pmatrix}. \quad (2.73)$$

zusammenfassen.

Dabei ist  $p$  die Anzahl der Merkmale und  $n$  die Anzahl der zur Verfügung stehenden Objekte. Die Merkmale und ihre Realisierungen sind also zeilenweise angeordnet, während die Objekte spaltenweise in die Datenmatrix geschrieben werden. Im Folgenden werden Vektoren darum immer als Spaltenvektoren geschrieben.

Für die Auswertung der Daten ist es üblich, den Erwartungswert  $E(\mathbf{X}) = \mu_\mathbf{X}$  und die Kovarianzmatrix  $\text{Cov}(\mathbf{X}, \mathbf{X}) = \Sigma_{\mathbf{X}\mathbf{X}}$  als grundlegende Größen heranzuziehen. Da diese Größen im Allgemeinen unbekannt sind, verwendet man in der Praxis stattdessen für den Erwartungswert den Schätzer des Mittelwertvektors  $\bar{\mathbf{x}}$  und anstelle der Kovarianzmatrix die Stichprobenkovarianzmatrix  $\mathbf{C}_{\mathbf{xx}}$ .

Konkret gilt

$$\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_p)^T = \frac{1}{n} \mathbf{D}_\mathbf{x} \mathbf{1}_n, \quad (2.74)$$

$$\mathbf{C}_{\mathbf{xx}} = \left( c_{ij} = \frac{1}{n} \sum_{r=1}^n (x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j) \right) = \frac{1}{n} (\mathbf{D}_\mathbf{x} - \bar{\mathbf{x}} \mathbf{1}_n^T)(\mathbf{D}_\mathbf{x} - \bar{\mathbf{x}} \mathbf{1}_n^T)^T. \quad (2.75)$$

Unter  $\mathbf{1}_n$  soll dabei ein Vektor der Länge  $n$  verstanden werden, dessen Einträge alle den Wert 1 zugewiesen bekommen.

Zudem sei an dieser Stelle eine um den Mittelwert bereinigte Datenmatrix definiert, die sogenannte zentrierte Datenmatrix

$$\widetilde{\mathbf{D}}_\mathbf{x} = \sqrt{\frac{1}{n-1}} (\mathbf{D}_\mathbf{x} - \bar{\mathbf{x}} \mathbf{1}_n^T). \quad (2.76)$$

Dadurch vereinfacht sich Gleichung 2.75 zu

$$\mathbf{C}_{\mathbf{xx}} = \widetilde{\mathbf{D}}_\mathbf{x} \widetilde{\mathbf{D}}_\mathbf{x}^T. \quad (2.77)$$

Diese Repräsentation der Eigenschaften als Dimensionen eines Vektorraumes ermöglicht die Transformation dieses Raumes. Eine besondere Art der Transformation sind sogenannte affine Abbildungen. Ihr Vorteil ist das gute Verständnis der zugrundeliegenden mathematischen Eigenschaften.

Im Folgenden seien  $\mathbf{X}$  der zu betrachtende Zufallsvektor,  $\mathbf{x}_r$  der Vektor der  $r$ -ten Spalte der Datenmatrix  $\mathbf{D}_\mathbf{x}$ ,  $\mathbf{Y}$  der transformierte Zufallsvektor,  $\mathbf{y}_r$  analog der Vektor der  $r$ -ten Spalte der Datenmatrix  $\mathbf{D}_\mathbf{y}$ ,  $\mathbf{A} \in \mathbb{R}^{q \times p}$  eine Transformationsmatrix und  $\mathbf{b} \in \mathbb{R}^q$  ein weiterer Vektor. Eine affine Abbildung wird damit als eine Funktion der Gestalt

$$\mathbf{Y} = \mathbf{f}(\mathbf{X}) = \mathbf{A}\mathbf{X} + \mathbf{b}, \quad (2.78)$$

$$\mathbf{D}_\mathbf{y} = \mathbf{A}\mathbf{D}_\mathbf{x} + \mathbf{b} \mathbf{1}_n^T$$

definiert.

Das besondere an affinen Abbildungen in der multivariaten Statistik sind ihre Auswirkungen auf den Erwartungswert

$$E(\mathbf{Y}) = E(\mathbf{A}\mathbf{X} + \mathbf{b}) = \mathbf{A}E(\mathbf{X}) + \mathbf{b}, \quad (2.79)$$



sowie den Mittelwert

$$\bar{\mathbf{y}} = \frac{1}{n} \sum_{r=1}^n \mathbf{y}_r = \frac{1}{n} \sum_{r=1}^n \mathbf{A} \mathbf{x}_r + \mathbf{b} = \mathbf{A} \frac{1}{n} \sum_{r=1}^n \mathbf{x}_r + \frac{1}{n} \sum_{r=1}^n \mathbf{b} = \mathbf{A} \bar{\mathbf{x}} + \mathbf{b}. \quad (2.80)$$

Für die Kovarianzmatrix gilt nach der Transformation

$$\begin{aligned} \Sigma_{\mathbf{Y}\mathbf{Y}} &= \text{Cov}(\mathbf{A}\mathbf{X} + \mathbf{b}, \mathbf{A}\mathbf{X} + \mathbf{b}), \\ &= \mathbf{E} \left( (\mathbf{A}\mathbf{X} + \mathbf{b} - \mathbf{E}(\mathbf{A}\mathbf{X} + \mathbf{b})) (\mathbf{A}\mathbf{X} + \mathbf{b} - \mathbf{E}(\mathbf{A}\mathbf{X} + \mathbf{b}))^T \right), \\ &= \mathbf{E} \left( (\mathbf{A}(\mathbf{X} - \mathbf{E}(\mathbf{X})) + \mathbf{b} - \mathbf{b}) (\mathbf{A}(\mathbf{X} - \mathbf{E}(\mathbf{X})) + \mathbf{b} - \mathbf{b})^T \right), \\ &= \mathbf{A} \mathbf{E} \left( (\mathbf{X} - \mathbf{E}(\mathbf{X})) (\mathbf{X} - \mathbf{E}(\mathbf{X}))^T \right) \mathbf{A}^T, \\ &= \mathbf{A} \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{A}^T. \end{aligned} \quad (2.81)$$

Analog folgt für die Stichprobenkovarianzmatrix

$$\begin{aligned} \mathbf{C}_{\mathbf{y}\mathbf{y}} &= \frac{1}{n-1} \sum_{r=1}^n (\mathbf{y}_r - \bar{\mathbf{y}})(\mathbf{y}_r - \bar{\mathbf{y}})^T, \\ &= \frac{1}{n} \sum_{r=1}^n (\mathbf{A}\mathbf{x}_r + \mathbf{b} - \mathbf{A}\bar{\mathbf{x}} - \mathbf{b})(\mathbf{A}\mathbf{x}_r + \mathbf{b} - \mathbf{A}\bar{\mathbf{x}} - \mathbf{b})^T, \\ &= \mathbf{A} \left( \frac{1}{n} \sum_{r=1}^n (\mathbf{x}_r - \bar{\mathbf{x}})(\mathbf{x}_r - \bar{\mathbf{x}})^T \right) \mathbf{A}^T, \\ &= \mathbf{A} \mathbf{C}_{\mathbf{x}\mathbf{x}} \mathbf{A}^T. \end{aligned} \quad (2.82)$$

Für die weitere Betrachtung sind affine Abbildungen interessant, wenn sie bestimmte Anforderungen bezüglich des Erwartungswertes und der Kovarianzmatrix erfüllen. Im Folgenden wird zu sehen sein, wie die Hauptkomponentenanalyse in Abschnitt 2.5.3 und die Multivariate Änderungsdetektion in Abschnitt 2.5.4 als affine Abbildungen definiert werden können.

Für die Berechnung sowohl der Mittelwertvektoren als auch der Stichprobenkovarianzmatrizen bei Verwendung von Gleitkommaarithmetik sind spezielle Methoden zur Summenbildung nötig, weil die Rundungsfehler bei direkter Summation zu groß werden. Diese Methoden sind in Chan u. a. (1979) dargestellt. Je nach Anwendungsfall wurden sie für die Implementierung affiner Abbildung im Rahmen dieser Arbeit eingesetzt.

### 2.5.3. Hauptkomponentenanalyse<sup>8</sup>

#### Theorie

Eine Möglichkeit der Transformation des Merkmalsraumes bietet die Hauptkomponentenanalyse (HKA), die auch unter dem englischen Namen *Principal Component Analysis* (PCA) bekannt ist. Die Idee der HKA ist es, die Gesamtvariabilität des Datensatzes oder zumindest einen großen Teil davon mithilfe einer begrenzten Anzahl untereinander unkorrelierter Variablen abzubilden. Diese Art der Merkmalsreduktion wird im Rahmen der Fernerkundung als Standardverfahren eingesetzt (siehe Richards 2013, 354ff.). Dadurch entstehen zwei Vorteile. Erstens kommen auf diese Weise die charakteristischen Eigenschaften der Objekte zum

<sup>8</sup>Die Grundlage für diesen Abschnitt bildet Listner (2008).

Vorschein. Eigenschaften, die bei allen betrachteten Objekten annähernd gleich sind, werden hingegen vernachlässigt. Zweitens bieten die unkorrelierten Merkmale des transformierten Raumes numerische Vorteile für die Multivariate Änderungsdetektion (siehe Abschnitt 2.5.4). Um eine hohe Gesamtvariabilität mit möglichst wenigen Variablen zu erreichen, kann eine Maximierung der Varianz der ersten Komponente des Bildvektors  $\mathbf{Y}$  als Ausgangspunkt gewählt werden. Formal kann dies unter Zuhilfenahme von 2.81 formuliert werden als

$$f(\mathbf{a}) = \mathbf{a}^T \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a} \rightarrow \max. \quad (2.83)$$

Durch Streckung des Vektors  $\mathbf{a}$  kann diese Funktion unbeschränkt wachsen, weshalb die Randbedingung

$$g(\mathbf{a}) = \mathbf{a}^T \mathbf{a} - 1 = 0 \quad (2.84)$$

gefordert wird. Durch diese Annahme wird zudem sichergestellt, dass der Merkmalsraum während der Optimierung nur rotiert und nicht gestreckt oder gestaucht wird.

Daraus lässt sich folgendes Optimierungsproblem mit Nebenbedingungen als Lagrange-Funktion  $L$  formulieren

$$L(\mathbf{a}) = \mathbf{a}^T \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a} - \lambda (\mathbf{a}^T \mathbf{a} - 1). \quad (2.85)$$

Mittels der Lagrange-Methode kann daraus wie folgt ein Eigenwertproblem abgeleitet werden.

$$\begin{aligned} L'(\mathbf{a}) &= 2\Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a} - 2\lambda \mathbf{a} = 0, \\ \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a} &= \lambda \mathbf{a}. \end{aligned} \quad (2.86)$$

Die Kovarianz von  $\mathbf{Y}$  wird somit in der ersten Komponente maximiert, wenn  $\mathbf{a} = \mathbf{z}_1$  der zum größten Eigenwert  $\lambda_1$  von  $\Sigma_{\mathbf{X}\mathbf{X}}$  gehörige Eigenvektor ist. Die zweite bis  $p$ -te Komponente von  $\mathbf{Y}$  ergibt sich analog durch eine Transformation mit den weiteren orthonormalen Eigenvektoren  $\mathbf{z}_2$  bis  $\mathbf{z}_p$  zu den absteigend geordneten Eigenwerten  $\lambda_2$  bis  $\lambda_i$ , wobei die Eigenwerte gemäß ihrer Vielfachheit mehrfach vorkommen können. Nach dem Satz von Courant-Fisher (siehe Horn und Johnson 1999, S. 179) gilt damit für die  $i$ -te Komponente des resultierenden Vektors  $\mathbf{Y}$ , dass in ihr die Varianz maximiert ist unter der Bedingung, dass diese Komponente mit den ersten  $i - 1$  Komponenten unkorreliert ist. Zusammengefasst wird durch die Hauptkomponentenanalyse ein Datensatz erzeugt, dessen einzelne Variablen paarweise unkorreliert sind und nacheinander in die Richtung der größten Varianz zeigen.

Auf Grundlage der vorangegangenen Überlegungen ist die Hauptkomponententransformation eine affine Abbildung der Gestalt

$$\mathbf{Y} = \mathbf{f}(\mathbf{X}) = \mathbf{Z}^T (\mathbf{X} - \mathbf{E}(\mathbf{X})). \quad (2.87)$$

Dabei gilt für  $\mathbf{Z}$ , dass

$$\Sigma_{\mathbf{X}\mathbf{X}} = \mathbf{Z} \text{diag}(\lambda_i) \mathbf{Z}^T = \mathbf{Z} \mathbf{\Lambda} \mathbf{Z}^T. \quad (2.88)$$

Die Gleichung (2.88) nennt man die Spektralzerlegung der Kovarianzmatrix  $\Sigma_{\mathbf{X}\mathbf{X}}$ .  $\mathbf{\Lambda}$  ist hierin die Diagonalmatrix der absteigend geordneten Eigenwerte von  $\Sigma_{\mathbf{X}\mathbf{X}}$  und  $\mathbf{Z}$  die Matrix der zugehörigen paarweise orthonormalen Eigenvektoren. Weil  $\Sigma_{\mathbf{X}\mathbf{X}}$  symmetrisch ist, kann eine solche Zerlegung immer erfolgen.

Auf die Stichprobe übertragen ersetzt man in Gleichung 2.88 die Kovarianzmatrix  $\Sigma_{\mathbf{X}\mathbf{X}}$  durch die Stichprobenkovarianzmatrix  $\mathbf{C}_{\mathbf{X}\mathbf{X}}$ , so dass gilt

$$\mathbf{C}_{\mathbf{X}\mathbf{X}} = \hat{\mathbf{Z}} \hat{\mathbf{\Lambda}} \hat{\mathbf{Z}}^T \quad (2.89)$$

und damit

$$\mathbf{D}_{\mathbf{y}} = \hat{\mathbf{Z}}^T (\mathbf{D}_{\mathbf{x}} - \bar{\mathbf{x}} \mathbf{1}_n^T) = \sqrt{n-1} \hat{\mathbf{Z}}^T \widetilde{\mathbf{D}}_{\mathbf{x}}. \quad (2.90)$$

Dabei wird in Gleichung 2.88 die Kovarianzmatrix  $\Sigma_{\mathbf{X}\mathbf{X}}$  durch die Stichprobenkovarianzmatrix  $\mathbf{C}_{\mathbf{X}\mathbf{X}}$  ersetzt.

Das besondere an der Hauptkomponententransformation ist, dass die entstehenden Bildvektoren den Erwartungswert  $\mathbf{0}$  haben und dass ihre Komponenten unkorreliert sind, d.h. die Nebendiagonaleinträge von  $\Sigma_{\mathbf{Y}\mathbf{Y}}$  sind gleich 0. Dies beweisen folgende Gleichungen unter Verwendung von (2.81) und (2.79):

$$\begin{aligned}\Sigma_{\mathbf{Y}\mathbf{Y}} &= \mathbf{A}\Sigma_{\mathbf{X}\mathbf{X}}\mathbf{A}^T = \mathbf{Z}^T\Sigma_{\mathbf{X}\mathbf{X}}\mathbf{Z} = \mathbf{Z}^T(\mathbf{Z}\mathbf{\Lambda}\mathbf{Z}^T)\mathbf{Z} = \mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p), \\ \mu_{\mathbf{Y}} &= \mathbf{A}\mathbf{E}(\mathbf{X}) + \mathbf{b} = \mathbf{Z}^T\mathbf{E}(\mathbf{X}) - \mathbf{Z}^T\mathbf{E}(\mathbf{X}) = \mathbf{0}.\end{aligned}\tag{2.91}$$

Analog gilt unter Verwendung von (2.82) und (2.80) für die Stichprobengrößen, dass

$$\begin{aligned}\mathbf{C}_{\mathbf{Y}\mathbf{Y}} &= \mathbf{A}\mathbf{C}_{\mathbf{X}\mathbf{X}}\mathbf{A}^T = \hat{\mathbf{Z}}^T\mathbf{C}_{\mathbf{X}\mathbf{X}}\hat{\mathbf{Z}} = \hat{\mathbf{Z}}^T(\hat{\mathbf{Z}}\hat{\mathbf{\Lambda}}\hat{\mathbf{Z}}^T)\hat{\mathbf{Z}} = \hat{\mathbf{\Lambda}} = \text{diag}(\hat{\lambda}_1, \dots, \hat{\lambda}_p), \\ \bar{\mathbf{y}} &= \mathbf{A}\bar{\mathbf{x}} + \mathbf{b} = \hat{\mathbf{Z}}^T\bar{\mathbf{x}} - \hat{\mathbf{Z}}^T\bar{\mathbf{x}} = \mathbf{0}.\end{aligned}\tag{2.92}$$

Da die Komponenten des Bildvektors  $\mathbf{Y}$  gemäß absteigender Varianz sortiert sind, ist die wesentliche Information des Datensatzes in den vorderen Komponenten zu vermuten. Anstelle alle Komponenten zu verwenden, sollten daher nur die Komponenten für die weitere Analyse eingesetzt werden, die einen Großteil der Gesamtvariabilität erklären. Mit  $w_i$  wird der Anteil der ersten  $i$  Hauptkomponenten an der Gesamtvariabilität definiert als

$$w_i = \frac{\sum_{j=1}^i \lambda_j}{\sum_{k=1}^p \lambda_k}.\tag{2.93}$$

Auf diese Art und Weise können jene Komponenten ausgewählt werden, die einen relevanten Anteil der Gesamtvariabilität gemäß einer Benutzerentscheidung, beispielsweise 95%, abdecken<sup>9</sup>. Der Vorteil dieses Verfahrens ist darüber hinaus, dass die Gesamtvariabilität rotationsinvariant ist und dadurch mit  $w_i$  auch derselbe Anteil der Gesamtvariabilität des Ausgangsdatsatzes erfasst wird.

Ein weiterer wichtiger Aspekt bei der Hauptkomponentenanalyse betrifft die Art der Komponenten des Datenvektors. Wenn die Einheiten oder sogar die physikalischen Größen, welche durch die Komponenten repräsentiert werden, innerhalb des Vektors unterschiedlich sind, so ist eine Normierung notwendig. Man verdeutlicht sich dies schnell, indem man den Effekt der Varianz einer einzelnen Komponente auf die Transformation betrachtet. Durch die Varianz einer Ausgangsvariable wird die Richtung dieser Ausgangsvariable gewichtet.

Wird stattdessen eine Normierung der Stichprobenvarianz auf den Wert 1 durch

$$\mathbf{D}_{\mathbf{x}'} = \text{diag}\left(\frac{1}{c_{ii}}\right)(\mathbf{D}_{\mathbf{x}} - \bar{\mathbf{x}}\mathbf{1}^T)\tag{2.94}$$

durchgeführt, so geht die allgemeine Transformationsgleichung 2.90 über in

$$\mathbf{D}_{\mathbf{y}'} = \hat{\mathbf{Z}}^T\mathbf{D}_{\mathbf{x}'} = \hat{\mathbf{Z}}^T\text{diag}\left(\frac{1}{c_{ii}}\right)(\mathbf{D}_{\mathbf{x}} - \bar{\mathbf{x}}\mathbf{1}^T).\tag{2.95}$$

Das bedeutet nichts anderes, als dass statt der Stichprobenkovarianzmatrix  $\mathbf{C}_{\mathbf{X}\mathbf{X}}$  die Stichprobenkorrelationsmatrix  $\mathbf{R}_{\mathbf{X}\mathbf{X}}$  als Ausgangspunkt der Transformation verwendet wird. Für

<sup>9</sup>Ob dieser Beitrag zur Gesamtvariabilität statistisch signifikant ist, kann über einen Test wie in Linting u. a. 2011 geprüft werden. Darüber hinaus wird in Mardia u. a. (1979, S. 235-236) ein Isotropietest vorgestellt, mit dem geprüft werden kann, ob die letzten  $k$  Eigenwerte gleich sind und somit die letzten  $k$  Hauptkomponenten ignoriert werden sollten. In dieser Arbeit wird auf die Nutzung dieser Tests verzichtet.

$\mathbf{C}_{\mathbf{x}'\mathbf{x}'}$  gilt dann

$$\begin{aligned}
 \mathbf{C}_{\mathbf{x}'\mathbf{x}'} &= \frac{1}{n-1} \mathbf{D}_{\mathbf{x}'} \mathbf{D}_{\mathbf{x}'}^T, \\
 &= \mathbf{diag} \left( \frac{1}{c_{ii}} \right) \frac{1}{n-1} (\mathbf{D}_{\mathbf{x}} - \bar{\mathbf{x}} \mathbf{1}^T) (\mathbf{D}_{\mathbf{x}} - \bar{\mathbf{x}} \mathbf{1}^T)^T \mathbf{diag} \left( \frac{1}{c_{ii}} \right), \\
 &= \mathbf{diag} \left( \frac{1}{c_{ii}} \right) \mathbf{C}_{\mathbf{x}\mathbf{x}} \mathbf{diag} \left( \frac{1}{c_{ii}} \right) = \mathbf{R}_{\mathbf{x}\mathbf{x}}, \\
 &= \mathbf{diag} \left( \frac{1}{c_{ii}} \right) \hat{\mathbf{Z}} \hat{\mathbf{\Lambda}} \hat{\mathbf{Z}}^T \mathbf{diag} \left( \frac{1}{c_{ii}} \right), \\
 &= \left( \frac{1}{c_{11}} \hat{\mathbf{z}}_1 \cdots \frac{1}{c_{pp}} \hat{\mathbf{z}}_p \right) \mathbf{diag} \left( \hat{\lambda}_1 \cdots \hat{\lambda}_p \right) \left( \frac{1}{c_{11}} \hat{\mathbf{z}}_1 \cdots \frac{1}{c_{pp}} \hat{\mathbf{z}}_p \right)^T, \\
 &= \sum_{i=1}^p \frac{\hat{\lambda}_i}{c_{ii}^2} \hat{\mathbf{z}}_i \hat{\mathbf{z}}_i^T, \\
 &= \hat{\mathbf{Z}} \mathbf{diag} \left( \frac{\hat{\lambda}_i}{c_{ii}^2} \right) \hat{\mathbf{Z}}^T.
 \end{aligned} \tag{2.96}$$

Für die Stichprobenkovarianzmatrix nach der Transformation ergibt sich, dass

$$\mathbf{C}_{\mathbf{y}'\mathbf{y}'} = \hat{\mathbf{Z}}^T \mathbf{C}_{\mathbf{x}'\mathbf{x}'} \hat{\mathbf{Z}} = \hat{\mathbf{Z}}^T \hat{\mathbf{Z}} \mathbf{diag} \left( \frac{\hat{\lambda}_i}{c_{ii}^2} \right) \hat{\mathbf{Z}}^T \hat{\mathbf{Z}} = \mathbf{diag} \left( \frac{\hat{\lambda}_i}{c_{ii}^2} \right). \tag{2.97}$$

Durch diese modifizierte Hauptkomponententransformation haben alle Komponenten des resultierenden Vektors den gleichen Maßstab. Alle Komponenten des Ausgangsvektors werden als gleichwertig angesehen. Es ist jedoch zu beachten, dass diese Gleichwertigkeit auch dazu führen kann, dass der Signal-Rausch-Abstand sinkt, wenn z.B. einige Kanäle der Eingabedaten hauptsächlich Rauschen und andere im wesentlichen nur Signal enthalten. Für die objekt-basierte Änderungsdetektion eignet sich eine Transformation auf Grundlage der Stichprobenkorrelation allerdings in besonderem Maße, weil sich die Maßstäbe der Eigenschaften in den Eingabedaten häufig stark unterscheiden. Als Beispiel ist die gleichzeitige Verwendung des Umfangs eines Objekts und der Standardabweichung innerhalb eines Spektralkanals in einem Datensatz zu nennen.

Im Rahmen dieser Arbeit kommt die Hauptkomponententransformation in zwei Bereichen zum Einsatz. Der erste Bereich ist die iterative Hauptkomponentenanalyse zur Änderungsdetektion. Der zweite ist die Vorbereitung der Daten für die Multivariate Änderungsdetektion (siehe Abschnitt 2.5.4).

Beim ersten Einsatzgebiet, der iterativen Hauptkomponentenanalyse, die u.a. in Wiemker (1997) für die pixel-basierte Änderungsdetektion verwendet wird, werden als Koordinatensystem zwei korrespondierende Objektmerkmale in den Datensätzen betrachtet. In diesem Koordinatensystem ordnen sich jene Objekte, denen keine Änderung widerfahren ist, entlang einer Geraden als schmale Punktwolke. Objekte mit Änderungen hingegen liegen abseits der Geraden. Die Hauptkomponententransformation legt die erste Hauptachse genau entlang der Geraden der unveränderten Objekte, da in diese Richtung die größte Varianz zu beobachten ist. Die zweite Hauptkomponente, welche orthogonal zur Geraden liegt, kann dadurch als Änderungsintensität interpretiert werden. Da für die Schätzung der Kovarianzmatrix auch die Änderungsobjekte verwendet wurden, muss in einem weiteren Schritt die Lage der ersten Hauptachse um deren Einfluss korrigiert werden. Dies kann erreicht werden, indem die

Verteilung der unveränderten Objekte in der zweiten Hauptachse als Normalverteilung modelliert wird. Die Annahme der Normalverteilung sollte bei der Anwendung des Verfahrens getestet werden. Für die korrigierte Kovarianz wird nun jedes Objekt des Bildes mit der Wahrscheinlichkeit gemäß dieser Verteilung gewichtet. Anschließend wird eine neue Hauptkomponententransformation erzeugt, deren erste Hauptachse weniger verfälscht entlang der Gerade der unveränderten Objekte liegt. Dieser Prozess wird solange wiederholt, bis sich die Lage der Hauptachsen nicht mehr ändert.

Der zweite Einsatzbereich, die Vorbereitung für die Multivariate Änderungsdetektion, umfasst die Anwendung der Hauptkomponententransformation der Eingangsdaten. Für die weitere Analyse werden dann die transformierten Daten verwendet. Dies hat den Vorteil, dass eine Merkmalsreduktion auf Basis von Gleichung 2.93 möglich ist und so nur relevante Informationen in die Änderungsdetektion einfließen. Darüber hinaus können bei der Multivariate Änderungsdetektion auftretende numerische Probleme durch die Entfernung von Redundanzen in den Daten behoben werden.

## Implementierung

Im Rahmen dieser Arbeit wurde die Hauptkomponententransformation als Plugin für *eCognition* implementiert. Auf der beiliegenden CD findet sich im Verzeichnis **PCA** das Plugin für verschiedene Versionen von *eCognition* einschließlich eines *eCognition*-Beispielprojekts für die Verwendung des Plugins.

Die im Plugin zu spezifizierenden Parameter für die Eingabedaten des Verfahrens bestehen erstens aus der Auswahl der zu verwendenden Objekteigenschaften (Parameter **Input Features**). Zweitens kann optional eine Wichtung der Objekte nach ihrer Größe erfolgen (Parameter **weighting by object size**). Des weiteren ist drittens eine Option vorhanden, die festlegt, ob die Korrelationsmatrix statt der Kovarianzmatrix als Ausgangspunkt für die Transformation benutzt werden soll (Parameter **use correlation matrix**). Viertens bietet das Plugin die Möglichkeit, eine bestehende Transformationsmatrix aus einer Datei zu laden (Parameter **Transformation parameters**). Schließlich kann fünftens eine Objektvariable spezifiziert werden, die festlegt, wie ein Objekt für die Schätzung der Transformationsmatrix gewichtet werden soll (Parameter **Object Weight Variable**).

Ausgabeparameter des Plugins sind einerseits die Objektvariablen, in die der jeweilige Wert nach der Hauptkomponententransformation abgelegt wird (Parameter **PCA component \***). Darüber hinaus können zwei Dateinamen angegeben werden. Die erste Datei dient als Speicherort für die Transformationsparameter (Parameter **Transformation parameters**). Bei der zweiten Datei handelt sich um eine CSV-Datei, welche die zu jeder Hauptachse gehörige Stichprobenvarianz angibt (Parameter **Variance**). Mithilfe der Informationen aus dieser Datei kann die Entfernung einzelner Hauptkomponenten auf Grundlage von Gleichung 2.93 realisiert werden.

Eine Übersicht der Parametereingabemaske findet sich in Abbildung 2.30.

Falls die Eingabe- und Ausgabedatei für die Speicherung der Transformationsparameter identisch sind, so wird diese Datei um die Daten aus dem aktuellen Bild erweitert. Dies ermöglicht die Anwendung der Hauptkomponentenanalyse im Rahmen der *Tiling-Stitching*-Methode analog zur Segmentierungsmethode aus Abschnitt 2.3.5. Dazu wird in einem ersten Durchlauf über alle Kacheln die Stichprobenkovarianzmatrix und der Stichprobenmittelwert ermittelt und gespeichert. Im zweiten Durchlauf wird die Hauptkomponententransformation mithilfe der gespeicherten Informationen auf jede Kachel angewandt. Für große Datenmengen, bei denen die Objektanzahl die verfügbaren Ressourcen überschreitet, wird so verhindert, dass alle Objekte gleichzeitig in den Speicher geladen werden müssen.

Abschließend sei auf ein Beispielergebnis des Plugins zur Berechnung Hauptkomponenten-

Parameter	Value
<input checked="" type="checkbox"/> Input PCA Features	
Input Features	[ Mean Layer 1, Mean Layer 2, Mean Layer 3 ] ...
weighting by object size	No
use correlation matrix	No
<input checked="" type="checkbox"/> Optional Input Components	
Transformation parameters	
Object Weight Variable	
<input checked="" type="checkbox"/> Output Components	
PCA component 1	pca01
PCA component 2	pca02
PCA component 3	pca03
<input checked="" type="checkbox"/> Optional Output Components	
Transformation parameters	transformation_parameters.it
Variance	variance_pca.csv

Abb. 2.30.: Eingabemaske des Plugins zur Hauptkomponentenanalyse für *eCognition*.

analyse in Abbildung 2.31 verwiesen. Dazu wurde das bereits in Abbildung 2.5 verwendete Bild mithilfe des *Multiresolution-Segmentation*-Algorithmus unter den Parametern  $T = 10$ ,  $w_s = 0.1$  und  $w_{comp} = 0.5$  segmentiert. Anschließend wurde das Plugin zur Hauptkomponententransformation auf die Objekt-Eigenschaften Mittelwert, Standardabweichung und Schiefe aller drei Ausgangskanäle angewandt. Die einzelnen dargestellten Bilder repräsentieren jeweils eine Hauptkomponente sortiert in absteigendem Anteil der Gesamtvarianz.

#### 2.5.4. Multivariate Änderungsdetektion<sup>10</sup>

##### Theorie

Die im letzten Abschnitt beschriebene Hauptkomponentenanalyse betrachtet jedes Objekt als Realisierung eines Zufallsvektors  $\mathbf{X}$ . Für die Änderungsdetektion ist jedoch eine Differenzierung zwischen den beiden Zeitpunkten sinnvoll. Somit liegt die direkte Modellierung der Beobachtungen als unterschiedliche Zufallsvektoren  $\mathbf{X}$  und  $\mathbf{Y}$  auf der Hand. Die Komponenten der beiden Zufallsvektoren sollten jedoch dieselben  $p$  Objekteigenschaften abbilden.

Eine Möglichkeit basierend auf diesem Modell, Unterschiede in den Daten hervorzuheben, bietet die Multivariate Änderungsdetektion (MAD) (siehe Nielsen u. a. 1998). Die Idee hinter diesem Verfahren ist es, die Vektoren  $\mathbf{X}$  und  $\mathbf{Y}$  so zu transformieren, dass die Varianz der Differenz der transformierten Vektoren maximiert wird. Diese Methode beruht auf der kanonischen Korrelationsanalyse (KKA), einem Verfahren der multivariaten Statistik. Bei der kanonischen Korrelationsanalyse wird jedoch im Gegensatz zur Multivariaten Änderungsdetektion die Varianz der Differenz minimiert.

Für die Erläuterung der Details des Verfahrens werden die positiv definiten Kovarianzmatrizen  $\Sigma_{\mathbf{X}\mathbf{X}}$ ,  $\Sigma_{\mathbf{X}\mathbf{Y}}$ ,  $\Sigma_{\mathbf{Y}\mathbf{Y}}$  und  $\Sigma_{\mathbf{Y}\mathbf{X}}$  definiert. Wie man sich leicht überlegt, gilt dann

$$\Sigma_{\mathbf{X}\mathbf{Y}} = \Sigma_{\mathbf{Y}\mathbf{X}}^T. \quad (2.98)$$

Zudem fordern wir ohne Beschränkung der Allgemeinheit, dass

$$\mathbf{E}(\mathbf{X}) = \mathbf{E}(\mathbf{Y}) = \mathbf{0}. \quad (2.99)$$

Die transformierten Merkmale der Segmente, deren Differenz die Varianz maximiert, werden auch als erste kanonische Variate  $U_1$  und  $V_1$  bezeichnet. Ihre Differenz  $M_1$  nennt man erste MAD-Variate.  $U_1$  und  $V_1$  lassen sich wie folgt definieren

$$\begin{aligned} U_1 &= \mathbf{a}_1^T \mathbf{X}, \\ V_1 &= \mathbf{b}_1^T \mathbf{Y}. \end{aligned} \quad (2.100)$$

<sup>10</sup>Die Grundlage für diesen Abschnitt bildet Listner (2008).



**(a)** Erste Hauptkomponente, repräsentiert 94% der Gesamtvarianz.



**(b)** Zweite Hauptkomponente, repräsentiert 3.0616% der Gesamtvarianz.



**(c)** Dritte Hauptkomponente, repräsentiert 1.5912% der Gesamtvarianz.



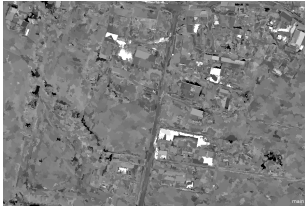
**(d)** Vierte Hauptkomponente, repräsentiert 0.8499% der Gesamtvarianz.



**(e)** Fünfte Hauptkomponente, repräsentiert 0.0774% der Gesamtvarianz.



**(f)** Sechste Hauptkomponente, repräsentiert 0.0436% der Gesamtvarianz.



**(g)** Siebte Hauptkomponente, repräsentiert 0.0083% der Gesamtvarianz.



**(h)** Achte Hauptkomponente, repräsentiert 0.0028% der Gesamtvarianz.



**(i)** Neunte Hauptkomponente, repräsentiert 0.018% der Gesamtvarianz.

**Abb. 2.31.:** Beispielergebnis des *eCognition*-Plugins zur Hauptkomponententransformation unter Verwendung der Objekt-Eigenschaften Mittelwert, Standardabweichung und Schiefe aller drei Farbkanäle des Bildes.

Für  $U_1$  und  $V_1$  gilt durch die lineare Transformation

$$\begin{aligned} E(U_1) &= \mathbf{a}_1^T E(\mathbf{X}) = 0, \\ E(V_1) &= \mathbf{b}_1^T E(\mathbf{Y}) = 0, \\ \text{Var}(U_1) &= \mathbf{a}_1^T \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a}_1, \\ \text{Var}(V_1) &= \mathbf{b}_1^T \Sigma_{\mathbf{Y}\mathbf{Y}} \mathbf{b}_1. \end{aligned} \tag{2.101}$$

Die Maximierung der Varianz kann somit geschrieben werden als Optimierungsaufgabe

$$M_1 = \text{Var}(U_1 - V_1) \rightarrow \max. \tag{2.102}$$

Damit die Lösung eindeutig bestimmt ist, wird als Nebenbedingung gefordert, dass die Varianz der transformierten Zufallsvariablen gleich 1 ist, d.h.

$$\text{Var}(U_1) = \text{Var}(V_1) = 1. \tag{2.103}$$

Damit kann die zu maximierende Varianz der Differenzen geschrieben werden als

$$\text{Var}(U_1 - V_1) = \text{Var}(U_1) + \text{Var}(V_1) - 2\text{Cov}(U_1, V_1) = 2(1 - \text{Corr}(U_1, V_1)). \tag{2.104}$$

Unter der Annahme einer positiven Korrelation zwischen  $U_1$  und  $V_1$ , entspricht die Varianzmaximierung der Differenz einer Minimierung der Korrelation zwischen den Segmenten der beiden Bilder unter den Nebenbedingungen in Gleichung 2.103.

Formal kann dies geschrieben werden als

$$\rho = \text{Corr}(U_1, V_1) = \frac{\text{Cov}(U_1, V_1)}{\sqrt{\text{Var}(U_1) \text{Var}(V_1)}} = \frac{\mathbf{a}_1^T \Sigma_{\mathbf{X}\mathbf{Y}} \mathbf{b}_1}{\sqrt{\mathbf{a}_1^T \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a}_1 \mathbf{b}_1^T \Sigma_{\mathbf{Y}\mathbf{Y}} \mathbf{b}_1}} = \text{Cov}(U_1, V_1) \rightarrow \min. \tag{2.105}$$

Mithilfe der Lagrange-Multiplikatoren kann diese Optimierungsaufgabe gelöst werden. Dazu wird die Lagrange-Funktion definiert als

$$\begin{aligned} L(\mathbf{a}_1, \mathbf{b}_1, \mu, \nu) &= \text{Cov}(U_1, V_1) - \frac{\nu}{2}(\text{Var}(U_1) - 1) - \frac{\mu}{2}(\text{Var}(V_1) - 1), \\ &= \mathbf{a}_1^T \Sigma_{\mathbf{X}\mathbf{Y}} \mathbf{b}_1 - \frac{\nu}{2}(\mathbf{a}_1^T \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a}_1 - 1) - \frac{\mu}{2}(\mathbf{b}_1^T \Sigma_{\mathbf{Y}\mathbf{Y}} \mathbf{b}_1 - 1). \end{aligned} \tag{2.106}$$

Diese wird nun nach den Parametern  $\mathbf{a}_1$  und  $\mathbf{b}_1$  sowie den Multiplikatoren  $\nu$  und  $\mu$  abgeleitet und 0 gesetzt. Damit ergibt sich

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{a}_1} &= \Sigma_{\mathbf{X}\mathbf{Y}} \mathbf{b}_1 - \frac{\nu}{2} 2 \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a}_1 = 0, \\ \frac{\partial L}{\partial \mathbf{b}_1} &= \Sigma_{\mathbf{X}\mathbf{Y}} \mathbf{a}_1 - \frac{\mu}{2} 2 \Sigma_{\mathbf{Y}\mathbf{Y}} \mathbf{b}_1 = 0, \\ \frac{\partial L}{\partial \nu} &= \mathbf{a}_1^T \Sigma_{\mathbf{X}\mathbf{X}} \mathbf{a}_1 - 1 = 0, \\ \frac{\partial L}{\partial \mu} &= \mathbf{b}_1^T \Sigma_{\mathbf{Y}\mathbf{Y}} \mathbf{b}_1 - 1 = 0. \end{aligned} \tag{2.107}$$

Es gilt somit

$$\begin{aligned} \mathbf{a}_1 &= \frac{1}{\nu} \Sigma_{\mathbf{X}\mathbf{X}}^{-1} \Sigma_{\mathbf{X}\mathbf{Y}} \mathbf{b}_1, \\ \mathbf{b}_1 &= \frac{1}{\mu} \Sigma_{\mathbf{Y}\mathbf{Y}}^{-1} \Sigma_{\mathbf{Y}\mathbf{X}} \mathbf{a}_1. \end{aligned} \tag{2.108}$$



Wird nun Gleichung 2.105 quadriert und die Ergebnisse aus 2.108 eingesetzt, so erhält man

$$\begin{aligned}\rho^2 &= \frac{\mathbf{a}_1^T \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} \mathbf{a}_1}{\mathbf{a}_1^T \Sigma_{XX} \mathbf{a}_1}, \\ \rho^2 &= \frac{\mathbf{b}_1^T \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} \mathbf{b}_1}{\mathbf{b}_1^T \Sigma_{XX} \mathbf{b}_1}.\end{aligned}\tag{2.109}$$

Weil  $\Sigma_{XX}$  und  $\Sigma_{YY}$  symmetrisch und positiv definit sind, existieren die Cholesky-Zerlegungen

$$\begin{aligned}\Sigma_{XX} &= \mathbf{L}_{XX} \mathbf{L}_{XX}^T, \\ \Sigma_{YY} &= \mathbf{L}_{YY} \mathbf{L}_{YY}^T.\end{aligned}\tag{2.110}$$

Mithilfe der Ersetzungen

$$\begin{aligned}\mathbf{a}_1 &= \mathbf{L}_{XX}^{-T} \widetilde{\mathbf{a}}_1, \\ \mathbf{b}_1 &= \mathbf{L}_{YY}^{-T} \widetilde{\mathbf{b}}_1,\end{aligned}\tag{2.111}$$

können die Gleichungen 2.109 geschrieben werden als

$$\begin{aligned}\rho^2 &= \frac{\widetilde{\mathbf{a}}_1^T \mathbf{L}_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} \mathbf{L}_{XX}^{-T} \widetilde{\mathbf{a}}_1}{\widetilde{\mathbf{a}}_1^T \widetilde{\mathbf{a}}_1} \\ \rho^2 &= \frac{\widetilde{\mathbf{b}}_1^T \mathbf{L}_{YY}^{-1} \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} \mathbf{L}_{YY}^{-T} \widetilde{\mathbf{b}}_1}{\widetilde{\mathbf{b}}_1^T \widetilde{\mathbf{b}}_1}.\end{aligned}\tag{2.112}$$

Diese sogenannten Rayleigh-Quotienten für  $\rho$  werden, Symmetrie der Matrix im Zähler vorausgesetzt, nach dem Satz von Rayleigh-Ritz (siehe Horn und Johnson 1999, S. 176) nicht kleiner als der kleinste Eigenwert. Gleichheit gilt dann und nur dann, wenn  $\widetilde{\mathbf{a}}_1$  bzw.  $\widetilde{\mathbf{b}}_1$  Eigenvektoren zum jeweils kleinsten Eigenwertes sind. Die Maximierung der Varianz ergibt sich somit durch die Transformation der Merkmalsvektoren mithilfe von  $\mathbf{a}_1$  und  $\mathbf{b}_1$ , die durch Eigenvektoren  $\widetilde{\mathbf{a}}_1$  und  $\widetilde{\mathbf{b}}_1$  zum jeweils kleinsten Eigenwert bestimmt sind. Dabei sollte jedoch beachtet werden, dass die Eigenvektoren nur bis auf ihre Vorzeichen eindeutig bestimmt sind. Um eine Minimierung der Kovarianz zu erreichen ist daher eine positive Korrelation zu fordern. Somit sind die Vorzeichen von  $\mathbf{a}_1$  und  $\mathbf{b}_1$  so zu wählen, dass gilt

$$\mathbf{a}_1^T \Sigma_{XY} \mathbf{b}_1 > 0.\tag{2.113}$$

Die Berechnung der ersten kanonischen Variate wurde somit auf ein Eigenwertproblem zurückgeführt. Interessant dabei ist, dass mithilfe des Satzes von Courant-Fisher (siehe Horn und Johnson 1999, S. 179) weitere kanonische Variate durch Transformationen  $\mathbf{a}_i$  und  $\mathbf{b}_i$  erzeugt werden können. Diese Transformationen maximieren unter der Bedingung, dass sie orthogonal sind zu allen  $\mathbf{a}_j$  und  $\mathbf{b}_j$  mit  $j < i$  sind, ebenfalls die Varianz der Differenz. Gegeben sind  $\mathbf{a}_i$  und  $\mathbf{b}_i$  durch die Eigenvektoren zu den weiteren Eigenwerten der Matrizen aus 2.112. Die Orthogonalität der Vektoren resultiert dabei direkt in paarweise unkorrelierten kanonischen Variaten, denn es gilt

$$\begin{aligned}\text{Cov}(U_i, U_j) &= \text{Cov}(\mathbf{a}_i^T \mathbf{x}, \mathbf{a}_j^T \mathbf{x}), \\ &= \text{Cov}(\widetilde{\mathbf{a}}_i^T \mathbf{L}_{XX}^{-1} \mathbf{X}, \widetilde{\mathbf{a}}_j^T \mathbf{L}_{XX}^{-1} \mathbf{X}), \\ &= \widetilde{\mathbf{a}}_i^T \mathbf{L}_{XX}^{-1} \Sigma_{XX} \mathbf{L}_{XX}^{-T} \widetilde{\mathbf{a}}_j, \\ &= \widetilde{\mathbf{a}}_i^T \mathbf{L}_{XX}^{-1} \mathbf{L}_{XX} \mathbf{L}_{XX}^T \mathbf{L}_{XX}^{-T} \widetilde{\mathbf{a}}_j, \\ &= \delta_{ij}.\end{aligned}\tag{2.114}$$

Für Paare von  $V_i$  und  $V_j$  folgt dies analog.  
Folglich ergeben sich auch weitere MAD-Variate  $M_i$  mit

$$M_i = U_i - V_i. \quad (2.115)$$

Der Vorteil dieser zusätzlichen Transformationen liegt darin, dass durch die Abwesenheit einer Korrelation zwischen den kanonischen Variaten auch die MAD-Variate unkorreliert sind (Canty 2009, S. 324). Dadurch werden verschiedene Arten von Änderungen in unterschiedlichen MAD-Variaten erfasst. Zur Klassifizierung von Änderungsarten ist dies hilfreich. Darüber hinaus lässt sich auf Grundlage der MAD-Variate eine Zufallsvariable  $Z$  definieren als

$$Z = \sum_{i=1}^p \frac{M_i^2}{\text{Var}(M_i)}. \quad (2.116)$$

Unter der Annahme, dass die  $M_i$  für unveränderte Objekte unkorreliert und normalverteilt<sup>11</sup> sind, so ist  $Z$  für die Klasse der unveränderten Objekte  $\chi^2$ -verteilt (siehe Canty 2009, S. 326). Mithilfe dieser Überlegungen ist es zudem analog zur iterierten Hauptkomponentenanalyse in Abschnitt 2.5.3 möglich, die Transformation um den Einfluss von Änderungsobjekten zu korrigieren. Aus diesem Grund wird die Methode auch als *Iteratively Reweighted Multivariate Alteration Detection* (IR-MAD) bezeichnet (Nielsen 2007).

Durch die in diesem Abschnitt beschriebene Methode der MAD bzw. IR-MAD werden zwei transformierte Merkmalsräume mit bis zu  $p$  Eigenschaften generiert, die sich durch minimale Korrelation zwischen beiden Zeitpunkten auszeichnen. Die Komponenten der Merkmalsräume sind zudem paarweise unkorreliert und ermöglichen so die geeignete Repräsentation unterschiedlicher Änderungstypen. Die Intensität der Änderung kann darüber hinaus über eine  $\chi^2$ -verteilte Zufallsgröße abgeleitet werden. Auf dieser Grundlage können relevante Änderungen bzw. verschiedene Änderungstypen mit Methoden aus Abschnitt 2.6 ermittelt werden.

## Implementierung

Um die Methode der IR-MAD im Rahmen der objekt-basierten Änderungsdetektion einsetzen zu können, wurde der Algorithmus im Rahmen dieser Arbeit als Plugin für *eCognition* implementiert. Als Grundlage dafür diente die bereits vorhandene Implementierung aus John (2009). Diese Implementierung musste jedoch aufgrund von teilweise unzureichender Programmierung und einiger weiterführender Anforderungen angepasst werden. Die Details dieser Anpassungen sollen im Folgenden erläutert werden.

Grund für die Anpassungen des Plugins aus John (2009) war die Tatsache, dass sich die Ergebnisse von denen des *ENVI*-Skripts aus Canty (2009, S. 331), das den IR-MAD-Algorithmus auf Pixelebene implementiert, erheblich unterscheiden. Grund dafür war die Normalisierung der einzelnen MAD-Variate auf das Intervall  $[0, 1]$ , was neben der Länge auch die Richtung der Vektoren und somit das Ergebnis verfälscht. Nachdem diese Normalisierung aus dem Programmcode entfernt wurde, lieferte das Plugin ein zu Canty (2009, S. 331) identisches Ergebnis.

Eine zweite Anpassung des Plugins aus John (2009) betrifft das numerische Verhalten des IR-MAD-Algorithmus. In Nielsen (2007) wird beschrieben, dass große Korrelationen zwischen den Komponenten von  $\mathbf{x}$  bzw.  $\mathbf{y}$  zu nahezu singulären Kovarianzmatrizen  $\Sigma_{\mathbf{xx}}$  bzw.  $\Sigma_{\mathbf{yy}}$  führen. Die Inversen dieser Matrizen werden jedoch bei der Berechnung der Transformation benötigt (siehe Gleichung 2.112). Lösen lässt sich das Problem auf zwei Arten. Eine Möglichkeit

<sup>11</sup>Diese Annahme muss bei der Anwendung statistisch getestet werden, ist aber wegen des zentralen Grenzwertsatzes und insbesondere bei vorheriger Anwendung der Hauptkomponentenanalyse aufgrund des zentralen Grenzwertsatzes plausibel (siehe Nielsen 2007, S. 465). Als Testverfahren sind beispielsweise der Kolmogorov-Smirnov-Test sowie der  $\chi^2$ -Anpassungstest geeignet. Für eine umfassende Übersicht zu statistischen Anpassungstests sei auf D'Agostino und Stephens (1986) verwiesen.

dazu bietet die Hauptkomponentenanalyse, mit deren Hilfe der Raum der Objektmerkmale auf unkorrelierte Merkmale reduziert werden kann, welche die wesentlichen Informationen des Datensatzes enthalten. Der dabei unvermeidlich auftretende Informationsverlust kann durch die Wahl der Hauptkomponenten entsprechend der minimal abzubildenden Gesamtvariabilität begrenzt werden (siehe Gleichung 2.93). Alternativ kann eine sogenannte Regularisierung der Matrizen durchgeführt werden, wie sie in Nielsen (2007) beschrieben wird. Im Rahmen dieser Arbeit wird jedoch die Methode zur Vorbereitung des Merkmalsraumes mithilfe der Hauptkomponentenanalyse gewählt. In Bezug auf das IR-MAD-Plugin musste dazu einerseits eine Nutzerausgabe im Falle einer Singularität implementiert werden, die angibt, nach welcher Iteration der Algorithmus instabil wird. Weiterhin wurde das Plugin dahingehend modifiziert, dass in einem solchen Fall die Transformation aus der letzten numerisch stabilen Iteration verwendet wird.

Drittens wurde die Möglichkeit geschaffen, die Transformationsdaten zu speichern und somit wie beim Plugin zur Durchführung der Hauptkomponentenanalyse eine Verarbeitung auf Kacheln zu ermöglichen. Die Funktionsweise wird über die beiden Algorithmen `Data Collection for IR-MAD-Transformation` und `Data Processing for IR-MAD-Transformation` realisiert.

Parameter	Value
Input MAD Features	
first component	[ PC1_time1, PC2_time1, PC3_time1 ] ...
second component	[ PC1_time2, PC2_time2, PC3_time2 ]
weighting by object size 1st option	No
weighting by object size 2nd option	No
Iterations	50
Output Components	
MAD component 1	mad1
MAD component 2	mad2
MAD component 3	mad3
Optional Output Components	
Chi^2 variable	chisquare
Output statistics	cca_variance.csv

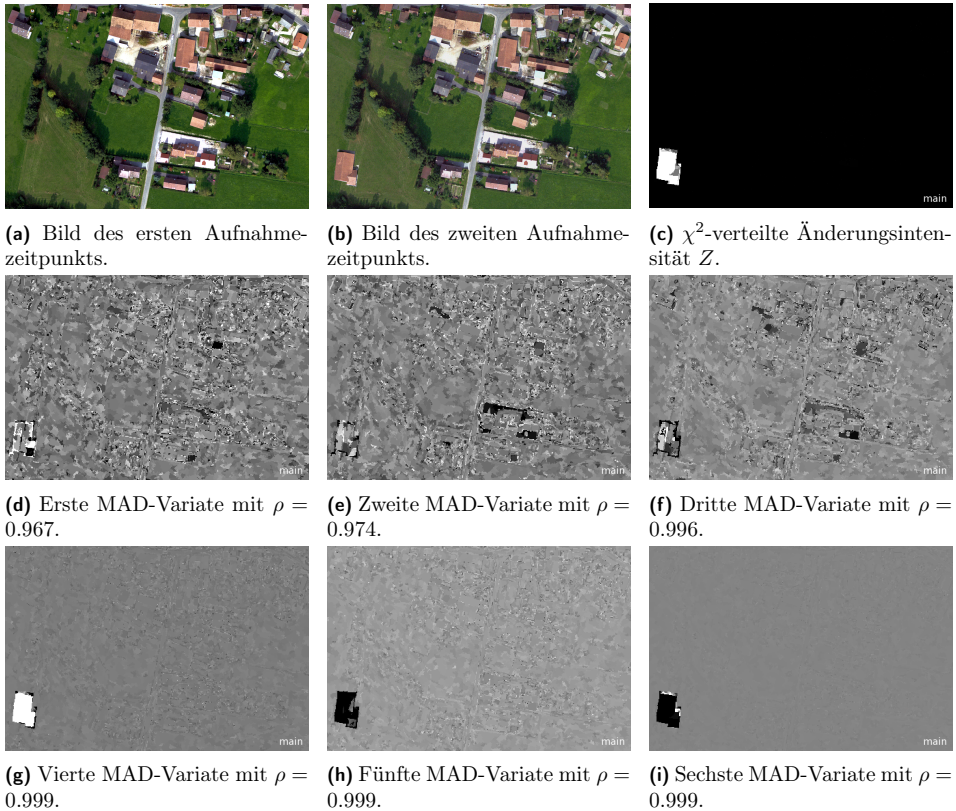
Abb. 2.32.: Eingabemaske des IR-MAD-Plugins für *eCognition*.

Um jedoch einen Überblick der Funktionsweise des Plugins zu geben, ist in Abbildung 2.32 die Eingabemaske der Parameter des Algorithmus abgebildet. Im Bereich **Input MAD Features** werden die Merkmale für den ersten Aufnahmezeitpunkt in Parameter **first component** und für den zweiten Aufnahmezeitpunkt in Parameter **second component** spezifiziert. Im Beispiel sind dies die ersten drei Hauptkomponenten der Datensätze der jeweiligen Aufnahmezeitpunkte. Mithilfe der Optionen im Parameter **weighting by object size 1st option** sowie dem Parameter **weighting by object size 2nd option** können die einzelnen Objekte bei der Schätzung des Mittelwerts und der Kovarianzmatrix unterschiedlich gewichtet werden (siehe dazu John (2009)). In diesem Beispiel wird darauf verzichtet, weshalb der Wert dieser Parameter auf „no“ gesetzt ist. Im Parameter **Iterations** wird festgelegt, wie viele Iterationen der Algorithmus höchstens durchlaufen soll. Das Beispiel verwendet den Standardwert von maximal 50 Iterationen. Erreicht der Algorithmus ein stabiles Ergebnis bereits mit weniger Iterationen, so bricht er vorzeitig ab.

Die Ausgabeparameter des Plugins werden in den Bereichen **Output Components** und **Optional Output Components** definiert. In den Parametern **MAD component \*** können dabei die Objektvariablen angegeben werden, welche den jeweiligen Wert der MAD-Variate zugewiesen bekommen, hier die Variable **mad1**, **mad2** und **mad3**. Das Ergebnis der  $\chi^2$ -Variable aus Gleichung 2.116 wird in der Objektvariable abgelegt, welche in Parameter **Chi^2 variable** spezifiziert wird. Im Beispiel ist dies die Variable **chisquare**. Schließlich kann im Parameter **Output statistics** eine CSV-Datei eingetragen werden, in die Informationen zur Korrelati-

on geschrieben werden. Im Beispiel wird dazu die Datei `cca_variance.csv` verwendet. Für die Algorithmen `Data Collection for IR-MAD-Transformation` und `Data Processing for IR-MAD-Transformation` sind zusätzlich die Parameter `Filename` bzw. `InputFile` verfügbar, mit denen die Transformationsdaten in einer Datei gespeichert, modifiziert und geladen werden können.

Das Plugin ist auf der beiliegenden CD einschließlich eines Beispiels im Verzeichnis `IR-MAD` abgelegt. Die Ergebnisse der Ausgabe dieses Beispiels finden sich zusätzlich in Abbildung 2.33.



**Abb. 2.33.:** Beispielergebnis des *eCognition*-Plugins zur Multivariaten Änderungsdetektion unter Verwendung der Objekt-Eigenschaften Mittelwert und Standardabweichung aller drei Farbkanäle.

### 2.5.5. Zusammenfassung

Im vorangegangenen Abschnitt wurden die Verfahren der HKA sowie der MAD in Hinblick auf ihre Anwendung im Bereich der objekt-basierten Änderungsdetektion präsentiert. Während die HKA eine maximale Varianzaufklärung des Gesamtdatensatzes anstrebt, wird bei der MAD versucht die Korrelation zwischen den beiden Datensätzen so gering wie möglich zu halten. Beide Verfahren sind für den Einsatz im Bereich der objekt-basierten Änderungsdetektion geeignet, da sie insbesondere mit unterschiedlichen Einheiten und Maßstäben innerhalb eines Datensatzes umgehen können. Die Multivariate Änderungsdetektion kann jedoch einen linearen Zusammenhang zwischen den beiden Datensätzen abbilden, den die Hauptkompo-

nentenanalyse nicht modellieren kann. Im Rahmen von Abschnitt 2.6.2 soll besonders auf die automatisierte Schwellenwertbestimmung bei nicht normalverteilten Merkmalen eingegangen werden.

Beide Verfahren wurden als Plugins für *eCognition* implementiert und hinsichtlich numerischer Stabilität optimiert. Die Anwendbarkeit der Verfahren auf große Bilder ist durch *Tiling-Stitching*-Methoden und Parallelisierung möglich. Hinsichtlich Ressourcenbedarf treten jedoch im Vergleich zur Bildsegmentierung auf modernen Rechnern kaum Engpässe auf.

Ein anwendungsorientierter Vergleich beider Verfahren wird in Abschnitt 3.3 dieser Arbeit gegeben.

## 2.6. Änderungsklassifizierung

### 2.6.1. Einleitung

Mit den Methoden des vorangegangenen Abschnitts 2.5 kann eine Repräsentation der Segmente in einem Merkmalsraum erstellt werden, die das Auffinden von Änderungsobjekten und sogar unterschiedlicher Arten von Veränderungen ermöglicht. Die so erzielte Repräsentation ist jedoch nicht zu verwechseln mit der Änderungsdetektion selbst. Denn die Entscheidung, ob sich ein Segment im Zeitraum zwischen den beiden Aufnahmezeitpunkten verändert hat, muss in einem nächsten Prozessschritt, der Änderungsklassifizierung, getroffen werden. Wie eine solche Änderungsklassifizierung realisiert werden kann, soll Inhalt dieses Abschnitts sein. Im Rahmen dieser Arbeit kommen drei verschiedene Gruppen von Klassifizierungsverfahren zum Einsatz. Diese Gruppierung ist nicht als abschließend zu betrachten. Für einen Überblick zu weiteren Klassifikationsverfahren sei jedoch auf Niemeyer (2001), Nussbaum und Menz (2008), Canty (2009) und Marpu (2009) verwiesen.

Die erste Art von Verfahren sind sogenannte Schwellenwertverfahren und werden in Abschnitt 2.6.2 präsentiert. Diese Methoden legen anhand von Ordnungsrelationen auf einem oder mehreren Merkmalen Grenzen für die einzelnen Klassen fest. Die Zugehörigkeit eines Objekts zu einer Klasse leitet sich dann aus den so definierten Merkmalsbereichen ab. Insbesondere das IR-MAD-Verfahren mit der  $\chi^2$ -verteilten Ergebnisgröße  $Z$  legt die Verwendung eines solchen Verfahrens nahe, um veränderte von unveränderten Bildbereichen zu trennen. Mithilfe von Verfahren zur automatischen Bestimmung des Schwellenwertes konnten unter anderem in Niemeyer (2001) und John (2009) gute Ergebnisse bei der Detektion von Änderungen erzielt werden. Eine komplexere Unterteilung in diverse Änderungsarten stößt jedoch mit schwellenwert-basierten Verfahren rasch an die Grenzen des Machbaren. Darüber hinaus eignen sich Schwellenwertverfahren auch zur Bestimmung des Segmentierungsparameters  $T_t$  aus Abschnitt 2.3.4.

Zur zweiten Gruppe, den sogenannten unüberwachten Klassifizierungsverfahren, gehören Algorithmen, die ohne Nutzerinteraktion die Objekte den Klassen zuordnen. Der sogenannte FMLE-Algorithmus ist ein Beispiel für diesen Verfahrenstyp. Dieser Algorithmus wurde zur Änderungsklassifizierung unter anderem in John (2009) und Canty (2009) erfolgreich eingesetzt. Daher lag für die Verwendung im Rahmen dieser Arbeit bereits eine Implementierung als *eCognition*-Plugin vor, welche im Rahmen dieser Arbeit benutzt und erweitert werden konnte. Da insbesondere das Problem der Wahl der richtigen Klassenanzahl in John (2009) ungelöst blieb, soll in Abschnitt 2.6.3 der Schwerpunkt auf dessen Lösung gelegt werden.

Die dritte Gruppe von Verfahren zur Änderungsklassifizierung bilden sogenannte überwachte Verfahren, welche in Abschnitt 2.6.4 vorgestellt werden. Diese Verfahren beruhen auf der Markierung von Trainingsobjekten für jede Klasse durch den Nutzer. Der Algorithmus klassifiziert die verbleibenden Objekte anschließend automatisch. Innerhalb dieser Gruppe werden exemplarisch zwei Verfahren herausgegriffen und ihre Einsatzmöglichkeiten für die Änderungsdetektion diskutiert. Da bisher der Fokus der Änderungsklassifizierung in vielen Arbeiten

(siehe z.B. John 2009) meist auf unüberwachten Verfahren lag, soll im Rahmen dieser Arbeit versucht werden, mithilfe überwachter Verfahren Änderungen zu klassifizieren und die dabei auftretenden Probleme zu analysieren.

### 2.6.2. Klassifizierung mit Schwellenwerten

Klassifizierungsverfahren mit Schwellenwerten werden in mehreren Arbeiten zur Bildanalyse in der Fernerkundung verwendet (siehe Niemeyer 2001; Bruzzone und Prieto 2002; Nussbaum u. a. 2006). Es ist daher nur folgerichtig, schwellenwert-basierte Verfahren auch im Rahmen dieser Arbeit für die objekt-basierte Änderungsdetektion zu benutzen. Allerdings wird in den genannten Arbeiten die Normalverteilung der Merkmale vorausgesetzt. Da in der Praxis und insbesondere bei der Änderungsdetektion diese Voraussetzung nicht immer gegeben ist, wird im Folgenden ein verallgemeinerter Ansatz vorgestellt, der prinzipiell auf die Normalverteilungsannahme verzichtet.

Die Grundidee von Klassifizierungsverfahren mit Schwellenwerten ist die Unterteilung der Achse der reellen Zahlen an einem definierten Wert  $x_t$ . Alle Objekte, für deren Realisierung des betreffenden Merkmals  $x \leq x_t$  gilt, werden dann der Klasse  $C_1$  und alle Objekte mit Werten  $x > x_t$  der Klasse  $C_2$  zugeordnet.

Zur Bestimmung der optimalen Wahl des Schwellenwertes  $x_t$  sei zunächst ein beliebiges Objektmerkmal als Zufallsgröße  $X$  durch folgende Wahrscheinlichkeitsdichte modelliert

$$f_X(x) = f_X(x|C_1)P(C_1) + f_X(x|C_2)P(C_2). \quad (2.117)$$

In der Praxis ist diese Verteilung nicht ohne weiteres gegeben und muss anhand von Trainingsbeispielen geschätzt werden.

Die Wahrscheinlichkeit einer Fehlklassifikation kann auf Grundlage von Gleichung 2.117 bestimmt werden durch

$$\begin{aligned} P_{MC}(x_t) &= \int_{-\infty}^{x_t} f_X(x|C_2)P(C_2) dx + \int_{x_t}^{\infty} f_X(x|C_1)P(C_1) dx, \\ &= F_X(x_t|C_2)P(C_2) + (1 - F_X(x_t|C_1))P(C_1). \end{aligned} \quad (2.118)$$

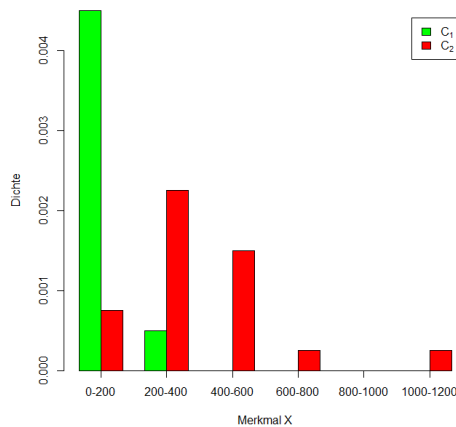
Zur Minimierung der Wahrscheinlichkeit einer Fehlklassifikation wird 2.118 nach  $x_t$  abgeleitet und gleich 0 gesetzt. Damit ergibt sich

$$f_X(x_t|C_1)P(C_1) = f_X(x_t|C_2)P(C_2) \quad (2.119)$$

als Bedingung für den optimalen Schwellenwert  $x_t$ . Sollte Gleichung 2.119 dabei mehrere Lösungen besitzen, so ist jene mit der geringsten Wahrscheinlichkeit für eine Fehlklassifikation zu verwenden.

Wenn  $f_X(\cdot|C_1)$  und  $f_X(\cdot|C_2)$  Dichtefunktionen von Normalverteilungen sind, so lässt sich auf diese Art analytisch eine optimale Lösung für  $x_t$  bestimmen (siehe Nussbaum u. a. 2006). Auch wenn beide Verteilungen einer logarithmischen Normalverteilung folgen, ist dies analog möglich. Im Allgemeinen kann für zwei beliebige Verteilungen jedoch keine analytische Lösung bestimmt werden. In diesem Fall sind numerische Verfahren, wie etwa das Newton-Verfahren (siehe z.B. Bronstein u. a. 2005), zur Lösung von Gleichung 2.119 einzusetzen.

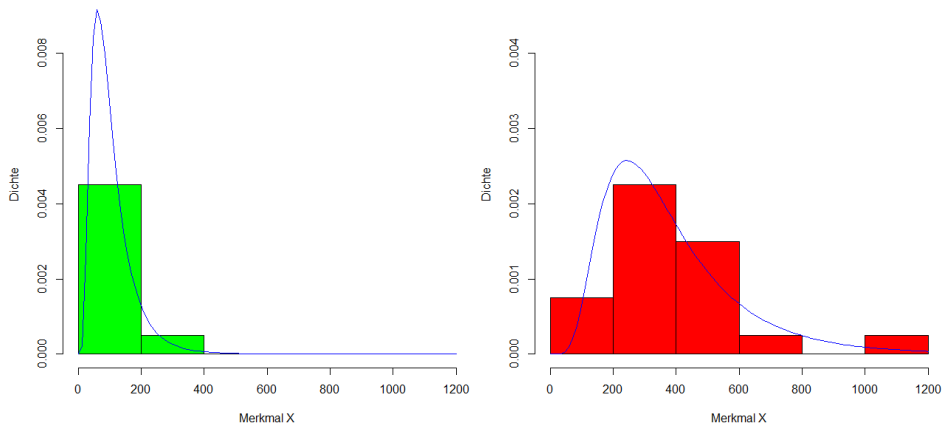
Um diese theoretische Darstellung des Verfahrens zu illustrieren, wurde die Methode als R-Skript implementiert. Auf der beiliegenden CD ist das Skript im Verzeichnis **ThresholdDetermination** als **analyze.r** abgelegt. Dort findet sich auch ein Beispieldatensatz in der Datei **dataset.csv**. In dem erwähnten Skript werden zunächst die Daten für die Klassen  $C_1$  und  $C_2$  geladen. Für



**Abb. 2.34.:** Gemeinsames Histogramm.

die Daten werden jeweils Mittelwert sowie Standardabweichung geschätzt und ein gemeinsames Histogramm erzeugt (siehe 2.34).

Anschließend wird mithilfe des Kolmogorov-Smirnov-Tests (K-S-Test) für jede der beiden Klassen aus Normalverteilung und logarithmischer Normalverteilung die am besten passende Dichtefunktion gewählt. Dazu werden die Werte der Teststatistik beider Dichten verglichen und jene Verteilung gewählt, die den niedrigsten Wert für die Teststatistik liefert, d.h. die den kleineren Fehler erster Art besitzt (siehe Abbildung 2.35a und 2.35b).



**(a)** Histogramm der Klasse  $C_1$  mit logarithmisch normalverteilter Dichte. **(b)** Histogramm der Klasse  $C_2$  mit logarithmisch normalverteilter Dichte.

Sind Form und Parameter der beiden Verteilungen ermittelt, kann Gleichung 2.119 numerisch gelöst werden und so der optimale Schwellenwert  $x_t$  bestimmt werden (siehe Abbildung 2.35).

Das beschriebene Verfahren zur Schwellenwertbestimmung ist prinzipiell auf beliebige Verteilungen erweiterbar, da der K-S-Test mit jeder Verteilungsfunktion kompatibel ist.

Vor einer Operationalisierung ist jedoch zu klären, ob der K-S-Test prinzipiell geeignet ist. Grund dafür ist die Tatsache, dass die Teststatistik des K-S-Tests bei Schätzung der Verteilungsparameter aus den Daten verfälscht wird. Alternativ könnte der Anderson-Darling-Test

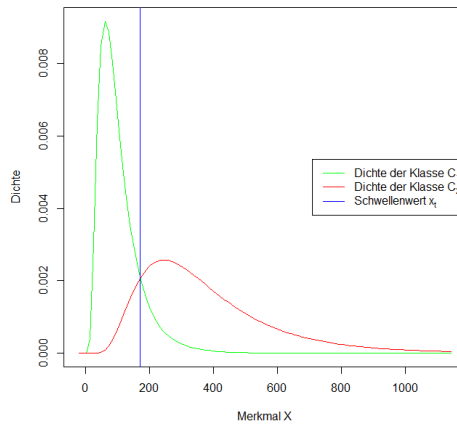


Abb. 2.35.: Schwellenwertbestimmung am Schnittpunkt der gewichteten Dichtefunktionen.

verwendet werden, der nicht durch dieses Phänomen betroffen ist (siehe D’Agostino und Stephens 1986).

Beim Einsatz in der Praxis ist zudem zu beachten, dass die a-priori Wahrscheinlichkeiten in Gleichung 2.117 geschätzt werden müssen. Daher sollte die Auswahl der für die Modellbildung verwendeten Trainingsbeispiele auf einer Gleichverteilung über den gesamten Bildbereich basieren.

Dass durch die hier vorgestellte Erweiterung der Methode *SEparability and THresholds* (SEaTH) aus Nussbaum und Menz (2008) eine Verbesserung der Klassifizierungsgenauigkeit erzielt werden kann, erscheint plausibel und wird in Abschnitt 3.3 empirisch belegt.

Neben der automatischen Bestimmung von Schwellenwerten zur Änderungsklassifizierung existiert auch die Möglichkeit, die Wahl des Schwellenwertes dem Nutzer zu überlassen. Dazu kann, wie in Abschnitt 2.3.4 beschrieben, die in *eCognition* vorhandenen Funktionen **Feature View** und **Image Object Information** genutzt werden. Ein Beispielergebnis für eine auf diese Weise erzielte Klassifikation auf Grundlage der Änderungsintensität  $Z$  (siehe Abschnitt 2.5.4) ist in Abbildung 2.36 zu sehen.

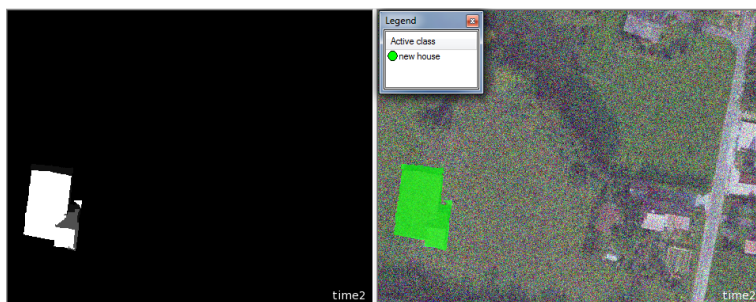


Abb. 2.36.: Beispielergebnis der interaktiven Schwellenwertklassifizierung (rechts) anhand des  $Z$ -Wertes (links).



### 2.6.3. Unüberwachte Klassifizierung mittels automatischem Clustering

Eine alternative Möglichkeit der Kategorisierung von Änderungen, die über eine simple Trennung zwischen unveränderten und veränderten Bildbereichen hinausgeht, bieten unüberwachte Klassifizierungsverfahren, sogenannte Clustering-Algorithmen. Methoden dieser Art werden als unüberwacht bezeichnet, da sie bis auf die transformierten Bilddaten und wenige Parameter keiner Nutzereingabe bedürfen. Anders ausgedrückt versuchen diese Verfahren, Strukturen nicht von Beispielen zu lernen, sondern durch die Daten selbst zu finden.

Konkret teilt die Clusteranalyse die Beobachtungen  $\mathbf{x}_1, \dots, \mathbf{x}_n$  im  $p$ -dimensionalen Merkmalsraum  $\mathbb{R}^p$  in  $c$  Klassen oder Kategorien mit den Mittelpunkten  $\mathbf{v}_1, \dots, \mathbf{v}_c$  ein. Dabei ist beabsichtigt, die Homogenität innerhalb eines Clusters zu maximieren, während zwischen unterschiedlichen Clustern größtmögliche Heterogenität herrschen sollte. Realisiert werden können diese Ziele durch den Einsatz einer Zielfunktion, welche dann mit mathematischen Methoden optimiert wird. Ein Beispiel für eine solche Funktion, die auch im weiteren Verlauf verwendet werden soll, findet sich in Gleichung 2.120.

$$Z(X; U, V) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^q \|x_j - v_i\|^2 \quad (2.120)$$

Der Wert  $u_{ij} \in [0, 1]$  beschreibt dabei die graduelle Zugehörigkeit des Objekts  $i$  zum Cluster  $j$  und der Exponent  $q \geq 1$  bestimmt den Grad der Unschärfe. Anstatt eines Algorithmus, bei dem zu jedem Verfahrensschritt eine feste Klasseneinteilung vorliegt, soll im Rahmen dieser Arbeit eine Zielfunktion verwendet werden, welche auf Fuzzy-Methoden beruht. Begründet wird diese Entscheidung mit den durch diesen Ansatz erzielten Ergebnissen in John (2009). Um die Zielfunktion zu optimieren, existiert eine Reihe von bestehende Verfahren z.B. der *Mean-Shift*-Algorithmus, das *K-Means*- und *Fuzzy-K-Means*-Verfahren, Agglomeratives Clustering sowie der FMLE-Algorithmus. Eine Übersicht dazu findet sich in Canty (2009, S. 267–309).

Beim Clustering treten laut Gath und Geva (1989) die folgenden drei Grundprobleme auf:

1. die Wahl der Anfangsschätzungen für die Clustermittelpunkte und damit verbunden die Stabilität der Ergebnisse und
2. die Variabilität in der Cluster-Form.
3. die „richtige“ Cluster-Anzahl  $c$  zu finden,

In Gath und Geva (1989) wird ein Verfahren vorgestellt, das diese Fragestellungen in den Clustering-Algorithmus einbezieht. Das Verfahren trägt den Titel *Fuzzy Maximum Likelihood Estimation* (FMLE). Es wurde zur Klassifizierung im Rahmen der Änderungsdetektion bereits in John (2009) eingesetzt und soll im Folgenden vorgestellt werden.

Der FMLE-Algorithmus beruht auf einer Variante des *K-Means*-Algorithmus aus Gath und Geva (1989). Dabei wird in einer ersten Phase des Verfahrens ausgehend von zufällig gewählten Clustermittelpunkten iterativ ein lokales Optimum der Zielfunktion 2.120 anhand des euklidischen Abstands gesucht. In der zweiten Phase wird das so gefundene Clustering als Ausgangspunkt für eine erneute Optimierung von 2.120 genutzt allerdings unter Verwendung einer exponentiellen Variante des Mahalanobisabstands.

Im Detail beginnt die erste Phase des FMLE-Algorithmus mit der zufälligen Wahl der initialen Clustermittelwerte  $v_1^{(0)}, \dots, v_c^{(0)} \in \mathbb{R}^p$  als Ausgangspunkt. Die Abstände zwischen den Beobachtungen  $x_1, \dots, x_n$  und den Clustermittelwerten werden anschließend und während der gesamten ersten Phase des Algorithmus über den quadratischen euklidischen Abstand

$$d^2(\mathbf{x}_i, \mathbf{v}_j) = (\mathbf{x}_i - \mathbf{v}_j)^T (\mathbf{x}_i - \mathbf{v}_j) \quad (2.121)$$

bestimmt, um die graduelle Zugehörigkeit jedes Segments zu jedem Cluster in der Partitionsmatrix

$$U = \left( u_{ij} = \frac{\left( \frac{1}{d^2(x_i, v_j)} \right)^{\frac{1}{q-1}}}{\sum_{k=1}^c \left( \frac{1}{d^2(x_i, v_k)} \right)^{\frac{1}{q-1}}} \right)_{i=1, \dots, n; j=1, \dots, c} \quad (2.122)$$

abbilden zu können.

Anschließend werden „verbesserte“ Clustermittelpunkte aus den Daten berechnet durch

$$v_j^{(m+1)} = \frac{\sum_{i=1}^n u_{ij}^q x_i}{\sum_{i=1}^n u_{ij}^q}. \quad (2.123)$$

Die beiden Schritte der Neuberechnung der Partitionsmatrix und die anschließende Verschiebung der Clustermittelpunkte werden solange wiederholt, bis keine wesentliche Verschiebung der Clustermittelpunkte mehr stattfindet, d.h bis zu einer Iteration  $m_{1,max}$  mit

$$\forall v_j^{(m_{1,max})} : \|v_j^{(m_{1,max}-1)} - v_j^{(m_{1,max})}\| \leq T_t. \quad (2.124)$$

Der Schwellenwert  $T_t$  wird dabei dem Verfahren als Parameter übergeben.

Die sich nun anschließende zweite Phase des FMLE-Algorithmus nutzt als Ausgangspunkt die Clustermittelpunkte  $v_1^{(m_{1,max})}, \dots, v_n^{(m_{1,max})}$  der vorangegangenen Phase. Im Anschluss wird erneut die Zugehörigkeitsmatrix über Gleichung 2.122 neu berechnet und die Clustermittelpunkte mit Gleichung 2.123 verschoben. Allerdings kommt dafür nicht mehr der euklidische Abstand aus Gleichung 2.121, sondern eine exponentielle Variante des Mahalanobis-Abstands mit

$$d_e^2(\mathbf{x}_i, \mathbf{v}_j) = \frac{\sqrt{|F_j|}}{p_j} e^{\frac{1}{2}(\mathbf{x}_i - \mathbf{v}_j)^T F_j^{-1} (\mathbf{x}_i - \mathbf{v}_j)}. \quad (2.125)$$

Dabei steht  $F_j$  für die geschätzte Kovarianzmatrix und  $p_j$  für die ebenfalls geschätzte A-priori-Wahrscheinlichkeit des Clusters  $j$ .

Prinzipiell entspricht die zweite Phase des Algorithmus einer Variante des *Expectation Maximization* (EM) (siehe Dempster u. a. 1977). Gath und Geva (1989) weist jedoch darauf hin, dass bei Verwendung des EM-Algorithmus mit zufällig initialisierten Klassenmittelpunkten die Stabilität der Ergebnisse in der Praxis nicht gesichert ist. Aus diesem Grund wird in der ersten Phase des FMLE-Algorithmus der *K-Means*-Algorithmus mit euklidischem Abstand für eine Vorauswahl der Klassenmittelpunkte genutzt, um das Grundproblem 1 zu behandeln. Ein großer Vorteil des Einsatzes des exponentiellen Mahalanobis-Abstandes  $d_e^2$  im Gegensatz zum euklidischen Abstand  $d_e$  besteht darin, dass die Cluster nicht zwingend hyperkugelförmig, sondern auch hyperellipsoid sein können. Damit wird das Grundproblem 2 zumindest teilweise gelöst.

Grundproblem 3 lässt sich mit dem FMLE-Algorithmus jedoch nicht ohne weiteres lösen, da die Clusteranzahl  $c$  ein Parameter des Verfahrens ist und dieser vom Nutzer bestimmt werden muss. Allerdings existieren quantitative Bewertungsmaße für die Güte eines Clusterings mit deren Hilfe ein geeigneter Wert für  $c$  gewählt werden kann. Dazu wird der FMLE-Algorithmus für verschiedene Werte für  $c$  ausgeführt und der zugehörige Wert des Gütemaßes berechnet. Anschließend wird jenes  $c$  bestimmt, für welches das Gütemaß optimal ist. Im Folgenden sollen zwei solcher Maße vorgestellt werden, die auch in Gath und Geva (1989) verwendet werden. Das erste Gütemaß wird *Fuzzy Hypervolume* genannt und kann über

$$F_{HV} = \sum_{i=1}^c \sqrt{|F_i|} \quad (2.126)$$

bestimmt werden.

Die Idee zur Konstruktion des *Fuzzy Hypervolume* beruht auf der wünschenswerten Eigenschaft eines Clusterings, dass die einzelnen Cluster eine möglichst geringe Ausbreitung im Merkmalsraum besitzen haben. Die Determinanten der Kovarianzmatrizen können über das Produkt der Längen der Eigenvektoren also die Varianzen in Hauptachsenrichtung berechnet werden. Je geringer diese Varianzen sind, desto kleiner ist auch die „Ausbreitung“ der Verteilung. Der optimale Wert für  $c$  kann daher anhand eines lokalen Minimums von 2.126 bestimmt werden.

Ein zweites Gütemaß kann definiert werden, indem die Zugehörigkeitswerte jener Objekte summiert werden, die innerhalb der Ein-Sigma-Umgebung der Clusterzentren liegen, und ins Verhältnis zum „Volumen“ des Clusters gesetzt werden. Mit

$$S = \sum_{i=1}^c \sum_{\mathbf{x}_j \in \sigma_1(C_i)} u_{ij}, \quad (2.127)$$

$$\sigma_k(C_i) = \{ \mathbf{x}_j | (\mathbf{x}_j - \mathbf{v}_i)^T F_i^{-1} (\mathbf{x}_j - \mathbf{v}_i) < k \},$$

berechnet man die *Partition Density* als

$$P_D = \frac{S}{F_{HV}}. \quad (2.128)$$

Motiviert wird die Größe  $P_D$  durch die Annahme, dass im Zähler die Masse der Clusterzentren und im Nenner das Volumen repräsentiert wird. In der Physik ist das Verhältnis von Masse zu Volumen als Dichte definiert, weshalb es hier gerechtfertigt erscheint, von der Dichte des Clusterings, der *Partition Density*, zu sprechen. Ein gutes Clustering zeichnet sich durch eine hohe Dichte in der Nähe der Cluster-Mittelpunkte aus. Daher kann ein geeigneter Wert für  $c$  mithilfe eines lokalen Maximums für  $P_D$  bestimmt werden.

Weitere Maße zur Bewertung der Güte eines Clusterings sind die *Average Partition Density* (siehe Gath und Geva 1989) und der Silhouettenindex (siehe Petrovic 2006). Diese beiden Gütemaße wurden zwar im Rahmen dieser Arbeit auch implementiert, sollen jedoch in den Anwendungsbeispielen in Kapitel 3 keine Rolle spielen.

Um das beschriebene Verfahren auch praktisch einsetzen zu können, wurde bereits in John (2009) ein Plugin für *eCognition* präsentiert, welches den FMLE-Algorithmus implementiert. Im Rahmen dieser Arbeit wurden die beiden beschriebenen Gütemaße im Plugin zusätzlich implementiert sowie eine Unzulänglichkeit in der Programmierung, die zum Absturz des Plugins führen konnte, behoben.

Parameter	Value
<input checked="" type="checkbox"/> Input features & parameter	
features	[mad1, mad2, mad3]
cluster	2
parameter m (fuzzifiers)	2
termination criterion	0.0001
result class group	cluster
Output statistics	fmle_output.csv

Abb. 2.37.: Ausschnitt der Eingabemaske des FMLE-Plugins für *eCognition*.

Die Eingabemaske des Plugins ist in Abbildung 2.37 dargestellt. Im Eingabebereich **features** werden die zum Clustering zu verwendenden  $p$  Objekteigenschaften festgelegt. Im Beispiel der Abbildung kommen hier die MAD-Variate aus der vorangegangenen Transformation zur Anwendung. Der Parameter **cluster** legt die Anzahl der zu verwendenden Cluster fest, im Beispiel 2, und ist somit gleichbedeutend zur in den obigen Formeln verwendeten Größe  $c$ .

In **parameter m (fuzzifiers)** wird der Grad der Unschärfe, also der Wert für  $m$  spezifiziert. Standardmäßig ist dieser Wert wie in Abbildung 2.37 auf den Wert 2 gesetzt. Das Abbruchkriterium  $T_t$  legt der Nutzer über den Parameter **termination criterion** fest. Das Beispiel in Abbildung 2.37 benutzt hier den Standardwert 0.0001. Um alle Cluster in *eCognition* einheitlich vorliegen zu haben, wird im Parameter **result class group** eine Klasse definiert, welche die einzelnen Cluster als Unterklassen enthalten soll. In Abbildung `methoden:fmle:pluginform:fig` wird die Klasse **cluster** verwendet. Schließlich kann optional in **Output statistics** eine CSV-Datei eingetragen werden, in welche die Werte für die Gütemaße exportiert werden. Sollte die Datei bereits existieren, so fügt der Algorithmus eine weitere Zeile an. Im Beispiel in Abbildung 2.37 kommt hierzu die Datei `fmle_output.csv` zum Einsatz.

Die Verwendung des Algorithmus zur Änderungsklassifizierung ist auf verschiedene Arten möglich. So kann einerseits ein Schwellwert für den  $\chi^2$ -Wert bestimmt werden, um nur relevante Änderungen zu betrachten. Anschließend wird der FMLE-Algorithmus auf den gefilterten Objekten unter Einsatz der MAD-Variate ausgeführt. Andererseits ist auch ein iteratives Clustering möglich, wobei in jedem Schritt der Nutzer jene Klasse entfernt, welche sicher keine Änderungen repräsentiert. Für beide Ansätze findet sich in Abschnitt 3.3.6 ein Anwendungsbeispiel.

#### 2.6.4. Überwachte Klassifizierung mit neuronalen Netzwerken

Einen Ansatz zur überwachten Klassifizierung, welcher im Rahmen der objekt-basierten Bildanalyse eingesetzt wird, bieten sogenannte neuronale Netzwerke. Unter anderem in Marpu (2009) wurde gezeigt, dass diese Verfahren instande sind, Klassifikationen mit hoher Güte zu liefern. Im Gegensatz zu schwellenwert-basierten Verfahren werden bei neuronalen Netzwerken zudem keine Verteilungsannahmen getroffen.

Neuronale Netzwerke imitieren die Funktionsweise des menschlichen Gehirns. Dazu werden Eingabeinformationen über sogenannte Neuronen in Ausgabeinformationen umgewandelt. Eingabeinformationen sind im Fall der objekt-basierten Bildanalyse die Merkmalsvektoren der Segmente. Die Ausgabe des neuronalen Netzes ist der Zugehörigkeitswert eines Segments zu einer zuvor definierten Klasse.

Praktisch erfolgt diese Umwandlung mithilfe mehrstufiger gewichteter Summen. Ein Klassifizierungsverfahren mit neuronalen Netzwerken ermittelt dazu in einem ersten Schritt, dem Modelltraining, anhand von Trainingsbeispielen und einem Trainingsalgorithmus für jede Klasse die Summengewichte des neuronalen Netzes. Im zweiten Schritt, dem Test des Modells, werden dem neuronalen Netz die Merkmalsvektoren bisher unbekannter Segmente zugeführt, um eine Klassifikation zu erhalten.

Eine spezielle Art von neuronalen Netzwerken, die im Rahmen von Marpu (2009) als Plugin für *eCognition* implementiert wurden, sind die *Class Based Neural Networks*. Dabei wird in einer ersten Stufe für jede verwendete Klasse ein eigenes Netzwerk erzeugt, wobei die Eingabeinformationen d.h. die verwendeten Merkmale je Klasse unterschiedlich sein können. Ausgabe der Netzwerke in der ersten Stufe sind Werte zwischen 0 und 1, die den Grad der Zugehörigkeit eines Segments zur jeweiligen Klasse repräsentieren. In der zweiten Stufe werden diese Zugehörigkeitswerte als Eingabeinformation eines weiteren neuronalen Netzwerkes verwendet, das die endgültige Klassifikation des Segments liefert. Da im Rahmen dieser Arbeit das Plugin *Class Based Neural Network* nicht weiterentwickelt wurde, sondern nur aus einer Anwendungsperspektive betrachtet wird, sei für weitere Details des Verfahrens auf Marpu (2009) verwiesen. Die Parametermaske des Plugins ist in Abbildung 2.38 zu sehen. Wie anhand dieses Eingabeformulars zu sehen ist, können im Bereich **Output Components** für jede verwendete Klasse unterschiedliche Eigenschaften definiert werden. Im Beispiel der Abbildung 2.38 werden jedoch für alle Klassen die MAD-Variate sowie der Z-Wert eingesetzt. Zu bemerken ist außerdem,

dass für die Parameter **Learning** der Wert **Mixed (Kalman & SCG)** die besten Ergebnisse liefert. Bezüglich der Anzahl verdeckter Ebenen des Netzwerks im Parameter **1 Hidden Layer** kann gesagt werden, dass dieser Parameter in der Kategorie **Class Network Setup** gleich der Anzahl der verwendeten Klassen sein sollte, hier 3. In der Kategorie **Classification Network Setup** sollte dieser Parameter jedoch genau um 1 größer gewählt werden.

Parameter	Value
Use class samples	no change, time 1 : car, time 2 : road, time 1 : road, time 2 : car
<b>Network Setup</b>	
<b>Class Network Setup</b>	
Learning	Mixed (Kalman+SCG)
1 Hidden Layer	3
Epochs	50
<b>Classification Network Setup</b>	
Learning	Mixed (Kalman+SCG)
1 Hidden Layer	4
Epochs	50
<b>Weights</b>	
Weights	No
<b>Log Options</b>	
Log Output	No
<b>Output Components</b>	
no change	[ chisquare, mad1, mad2, mad3 ]
time 1 : car, time 2 : road	[ chisquare, mad1, mad2, mad3 ]
time 1 : road, time 2 : car	[ chisquare, mad1, mad2, mad3 ]

**Abb. 2.38.:** Eingabemaske des *eCognition*-Plugins zur Klassifizierung mit neuronalen Netzwerken.

Neben dem Vorteil der hohen Qualität der Ergebnisse hat das Verfahren der Klassifizierung mit neuronalen Netzwerken insbesondere für die Änderungsklassifizierung einen wesentlichen Nachteil. Dieser Nachteil besteht in der Voraussetzung, dass alle Klassen vor der Klassifikation bekannt sein müssen. Zudem sind für jede Klasse einige Beispielsegmente vom Nutzer auszuwählen. Zwar ist laut Marpu (2009) nur eine sehr geringe Anzahl an Trainingsbeispielen nötig, jedoch ergibt sich die Menge der Klassen und somit auch die Auswahl der Beispiele erst aus den Daten. Eine vorherige manuelle Analyse des Datensatzes wäre daher nötig, was jedoch der Intention einer automatisierten Änderungsdetektion widerspricht.

Trotz dieser Nachteile ist das Verfahren der Änderungsklassifizierung mit neuronalen Netzwerken in einigen Einsatzbereichen ein adäquates Mittel. Im Rahmen dieser Arbeit wird das Verfahren daher in Abschnitt 3.2 eingesetzt. Dort findet sich auch eine Genauigkeitsuntersuchung zur Abschätzung der Klassifizierungsgüte.

### 2.6.5. Zusammenfassung

In diesem Abschnitt konnten drei verschiedene Arten der Änderungsklassifizierung präsentiert werden. Alle drei Methoden besitzen unterschiedliche Vorzüge, so dass die Wahl des zu verwendenden Verfahrens in Abhängigkeit des Anwendungsfalls getroffen werden sollte.

Der erste Methodentyp, die Klassifizierung mit Schwellenwerten, ermöglicht insbesondere die Unterteilung zwischen veränderten und unveränderten Gebieten. Dazu konnte in diesem Abschnitt ein Verfahren zur Bestimmung der Schwellenwerte aufgezeigt werden, das auf die Normalverteilungsannahme verzichtet.

Die zweite hier vorgestellte Methode besteht in der unüberwachten Klassifizierung der Änderungen mithilfe des FMLE-Algorithmus. Da hierzu keine Trainingsgebiete gewählt werden müssen, bietet dieses Verfahren einen hohen Grad an Automatisierung. Ein Parameter des Algorithmus, der in der Praxis häufig mithilfe der Methode „Versuch und Irrtum“ gewählt wird, ist die Anzahl der zu verwendenden Klassen. In diesem Zusammenhang konnte im vorliegenden Abschnitt eine Methode präsentiert werden, die es ermöglicht, zwei Klassifikationen des Algorithmus mit unterschiedlicher Anzahl an Klassen miteinander hinsichtlich ihrer Anpassung an die Bilddaten zu vergleichen.

Abschließend wurde ein Verfahren der überwachten Klassifizierung, die neuronalen Netzwer-

ke, präsentiert. Dieser Ansatz eignet sich besonders in dem Fall, dass sich die Änderungsarten aus der zu bearbeitenden Szene bzw. der Fragestellung ergeben. Während des Prozesses ist nämlich einerseits die Definition der Klassen als auch die Bereitstellung von Trainingsgebieten nötig. In realen Anwendungen zur Änderungsdetektion mag dies ein Hinderungsgrund für die Verwendung dieses Verfahrens darstellen.

## 2.7. Nachverarbeitung

### 2.7.1. Einleitung

Im Anschluss an die Änderungsdetektion sind weitere Schritte nötig, um dem Analyseergebnis seine abschließende Form zu geben. Diese Nachverarbeitungsschritte umfassen nach Stow (2010) drei Aspekte, die im Folgenden kurz erläutert werden sollen.

Den ersten Schritt bildet die Kartengeneralisierung und Bearbeitung. Darin werden zum einen benachbarte Objekte derselben Klasse miteinander Verbunden. Zum anderen werden Objekte kleiner Ausdehnung entfernt, da diese häufig Artefakte des Prozesses sind. Um insbesondere Effekte durch Registrierungsfehler zu beheben, können Formeigenschaften verwendet werden, da sich diese Fehler häufig in schmalen, länglichen Objekten manifestieren. Auch werden im ersten Schritt die erzeugten extrahierten Änderungsobjekte in ihrer äußeren Form vereinfacht.

Im zweiten Schritt, der Änderungsberichtigung, werden bei klassifikations-basierten Änderungsdetektionsmethoden unplausible Klassenübergänge aus dem Analyseergebnis ausgeschlossen. Das Verfahren in Strahler u. a. (1980) bietet eine Möglichkeit dieses Ziel zu erreichen, indem Änderungen in der Klassifikation mithilfe von Markov-Ketten modelliert und so Wahrscheinlichkeiten für einen bestimmten Übergang zwischen zwei Klassen bestimmt werden können. Diese Wahrscheinlichkeitswerte finden dann als a-priori Wahrscheinlichkeiten Eingang in eine Klassifikation mittels Bayes-Regel.

Der dritte Schritt laut Stow (2010) besteht im manuellen Bearbeiten des Ergebnisses durch den Analysten. An dieser Stelle des Prozesses kann versucht werden, ausgehend von der Betrachtung einzelner Objekte mithilfe von *eCognition* Regeln aufzustellen, die eine solche Bearbeitung vereinfachen.

Die Methoden der Nachverarbeitung können im Rahmen dieser Arbeit nicht in allen Details betrachtet werden. Exemplarisch soll jedoch der Prozess der Kurvenglättung aus dem Bereich der Kartengeneralisierung im folgenden Unterabschnitt untersucht werden. Für weitere Informationen zur Nachverarbeitung sei neben Stow (2010) auch auf Niemeyer (2001) verwiesen.

### 2.7.2. Datenreduktion durch Kurvenglättung der Änderungsobjekte

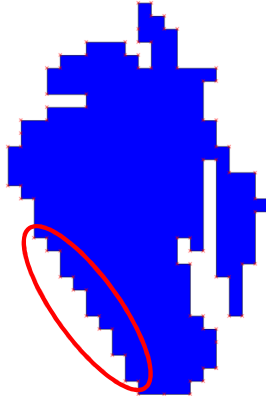
Wie bei jeder kartographischen Aufgabe ist auch bei der Kartierung von Änderungen die Generalisierung von Objekten ein wichtiger Aspekt, um eine verständliche und gut lesbare Karte zu erhalten, die auf das verwendete Medium und den eingesetzten Maßstab angepasst ist. Unter Generalisierung versteht man dabei die Vereinfachung von Kartenobjekten, die im gegebenen Maßstab nicht mehr vollständig dargestellt werden können. Diese Tätigkeit wird traditionell von einem Kartographen durchgeführt. Durch den immer weiter fortschreitenden Einsatz von Rechnern bei der Darstellung von geografischen Informationen steigt jedoch der Bedarf an automatisierten Verfahren zur Generalisierung.

Ein wesentlicher Teilbereich der Generalisierung ist die vereinfachte Darstellung von Begrenzungslinien<sup>12</sup>. In Leu und Chen (1988) wurden dazu Anforderungen an einen Algorithmus zur Generalisierung von Linien formuliert: Erstens soll ein Objekt unter weitgehender Beibehaltung seiner charakteristischen Form generalisiert werden. Zweitens darf die Vereinfachung

<sup>12</sup>Eine Übersicht weiterer Generalisierungsoperationen findet sich in Lee (1996).

der Form keinen übermäßig großen Rechenaufwand erzeugen und die neue Repräsentation möglichst wenig Speicher benötigen. Drittens wird von dem Algorithmus gefordert, dass die erzeugte Form stabil ist gegenüber Störungen durch Translationen, Rotationen und Blickwinkelveränderungen sowie durch Rauschen und partielle Verdeckung.

Darüber hinaus entstehen während der Verarbeitung im Rahmen der objekt-basierten Änderungsdetektion Artefakte an den Außenkanten der Objekte, die auf die Rasterung der Daten zurückzuführen sind (siehe Abbildung 2.39) und durch eine Repräsentation der Begrenzungslinien mit weniger Stützpunkten entfernt werden können.



**Abb. 2.39.:** Beispiel einer Linie, deren Digitalisierung zu viele Punkte aufweist.

Um das Problem der Generalisierung automatisiert zu lösen, ist der Ansatz, nur jeden  $k$ -ten Punkt der Ausgangslinie in die generalisierte Linie zu übernehmen, die einfachste Möglichkeit. Dieses Verfahren ist aus anderen Bereichen der Signalverarbeitung als reguläre Unterabtastung bekannt. Es entfernt unabhängig von der Form einer Linie eine festgelegte Anzahl an Punkten aus dem Polygonzug. Dadurch ergibt sich der Vorteil eines festen Reduktionsfaktors,  $\frac{1}{k}$ , sowie einer linearen Laufzeit des Algorithmus. Andererseits nutzt der skizzierte Algorithmus in keiner Weise, wie sich die Lage der Punkte im Ausgangspolygon zueinander verhält. Liegen etwa alle Punkte eines Polygonzugs auf einer Geraden, erzeugt der Algorithmus eine redundante Repräsentation. Auf der anderen Seite kann der Algorithmus nicht garantieren, dass Punkte, welche die Form des Polygonzugs maßgeblich beeinflussen, erhalten werden.

Ein etwas aufwändigerer Algorithmus, der die genannten Nachteile des Verfahrens der regulären Unterabtastung nicht hat, ist der Douglas-Peucker-Algorithmus aus Douglas und Peucker (1973)<sup>13</sup>. Da dieser Algorithmus die gesamte Entwicklung der computergestützten Kartographie beeinflusst hat, soll im Folgenden das Verfahren im Detail erläutert werden. Anschließend werden Eigenschaften des Algorithmus sowie einige Modifikationen bzw. Erweiterungen besprochen.

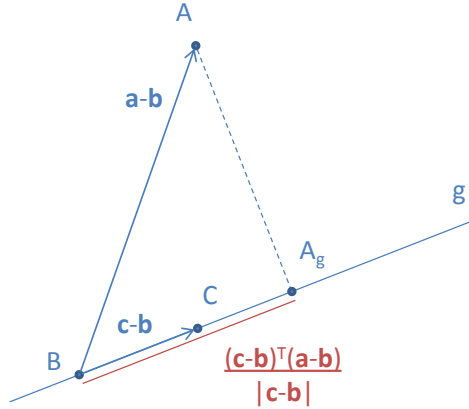
Sei  $P = (P_1, \dots, P_n)$  der zu glättende Polygonzug. Weiterhin sei  $d(A, \overline{BC})$  der euklidische Abstand zwischen dem Punkt  $A$  und der Strecke  $\overline{BC}$ . Ausgehend davon sucht der Algorithmus von Douglas und Peucker nach dem Punkt  $P_m$  zwischen  $P_1$  und  $P_n$ , der den größten Abstand zur Strecke  $\overline{P_1P_n}$  hat, d.h.

$$m = \arg \max_{i=2, \dots, n-1} d\{P_i, \overline{P_1P_n}\}. \quad (2.129)$$

Falls der maximale Abstand  $d(P_m, \overline{P_1P_n})$  kleiner gleich einem gegebenem Maximalwert  $\epsilon$  ist, so können alle Punkte verworfen werden, die zwischen  $P_1$  und  $P_n$  liegen, da sie nicht

<sup>13</sup>Der Algorithmus wurde im Bereich der Computergraphik unabhängig und parallel zu den Entwicklungen in der Kartographie als Ramer-Algorithmus bekannt (siehe Ramer 1972).

wesentlich von der Strecke  $P_1, \dots, P_n$  abweichen und durch die Strecke selbst gut repräsentiert werden. Andernfalls sollte zumindest der Punkt  $P_m$  nicht verworfen werden. Anschließend führt man rekursiv den Algorithmus jeweils voneinander unabhängig für die Polygonzüge  $P_L = (P_1, \dots, P_m)$  und  $P_R = (P_m, \dots, P_n)$  durch.



**Abb. 2.40.:** Ermittlung des Lotfußpunktes  $A_g$  beim Fällen des Lotes von Punkt  $A$  auf die Gerade  $g$ .

Ein Aspekt, der bei der Beschreibung des Algorithmus bisher offen blieb, ist die Berechnung des euklidischen Abstands zwischen einem Punkt  $A$  und einer Strecke  $\overline{BC}$ . Hierfür betrachte man die Gerade  $g$  durch die Punkte  $B$  und  $C$

$$g : \mathbf{x}(t) = \mathbf{b} + t(\mathbf{c} - \mathbf{b}) \quad (2.130)$$

wie in Abbildung 2.40 dargestellt. Fällt man nun das Lot von Punkt  $A$  auf die Gerade  $g$  mit dem Lotfußpunkt  $A_g$ , so ist der Abstand  $d(A, A_g)$  die Länge der kürzesten Strecke zwischen  $A$  und  $g$ .<sup>14</sup> Der Lotfußpunkt  $A_g$  ergibt sich aus

$$\mathbf{a}_g = \mathbf{b} + t(\mathbf{c} - \mathbf{b}), \quad (2.131)$$

denn zum Punkt  $A_g$  gelangt man, indem zum Vektor  $\mathbf{b}$  ein skales Vielfaches des Vektors  $\mathbf{c} - \mathbf{b}$  addiert wird. Der Skalar  $t$  kann dabei aus dem Verhältnis der Länge der Strecke  $\overline{BA_g}$  zur Länge des Vektors  $\mathbf{c} - \mathbf{b}$  berechnet werden. Es gilt

$$\overline{BA_g} = \cos \angle (\overline{BA_g}, \overline{BC}) |\mathbf{a} - \mathbf{b}|, \quad (2.132)$$

und unter Ausnutzung der Definition des Skalarproduktes

$$(\mathbf{c} - \mathbf{b})^T (\mathbf{a} - \mathbf{b}) = \cos \angle (\overline{BA_g}, \overline{BC}) |\mathbf{c} - \mathbf{b}| |\mathbf{a} - \mathbf{b}|, \quad (2.133)$$

folgt

$$t = \frac{\overline{BA_g}}{\overline{BC}} = \frac{(\mathbf{c} - \mathbf{b})^T (\mathbf{a} - \mathbf{b})}{|\mathbf{c} - \mathbf{b}|^2}. \quad (2.134)$$

Für  $t \in [0, 1]$  liegt  $A_g$  auf der Strecke  $\overline{BC}$  und somit ist der gesuchte Abstand gleich jenem zwischen  $A$  und der Gerade  $g$ . Ist jedoch  $t \notin [0, 1]$  so ist der gesuchte Abstand über einen

<sup>14</sup>Diesen Abstand kann man sehr einfach durch Einsetzen des Ortsvektors  $\mathbf{a}$  in die Hessesche Normalform von  $g$  berechnen. Für die weiteren Betrachtungen ist dieser Ansatz jedoch nicht ausreichend.



der Endpunkte der Strecke zu ermitteln. Wird dies nicht beachtet und weiterhin der Abstand  $d(A, A_g)$  verwendet, wie ursprünglich in Douglas und Peucker (1973) beschrieben, so werden Punkte verworfen, die einen großen Abstand zur Strecke  $\overline{P_1 P_n}$  und somit auch einen wesentlichen Beitrag zur Form des Polygons haben können. Dieser Aspekt des Algorithmus wurde bereits in Ebisch (2002) erkannt und behoben. Zusammengefasst kann der Abstand zwischen dem Punkt  $A$  und der Strecke  $\overline{BC}$  wie folgt berechnet werden:

$$d(A, \overline{BC}) = \begin{cases} |\mathbf{b} - t(\mathbf{c} - \mathbf{b}) - \mathbf{a}| & \text{wenn } t \in [0, 1], \\ |\mathbf{b} - \mathbf{a}| & \text{wenn } t < 0, \\ |\mathbf{c} - \mathbf{a}| & \text{wenn } t > 1. \end{cases} \quad (2.135)$$

Um die Funktionsweise des Algorithmus beispielhaft zu verdeutlichen, werden in Abbildung 2.41 die Ergebnisse der Anwendung des Algorithmus auf ein Polygon mit verschiedenen Werten für  $\varepsilon$  dargestellt.

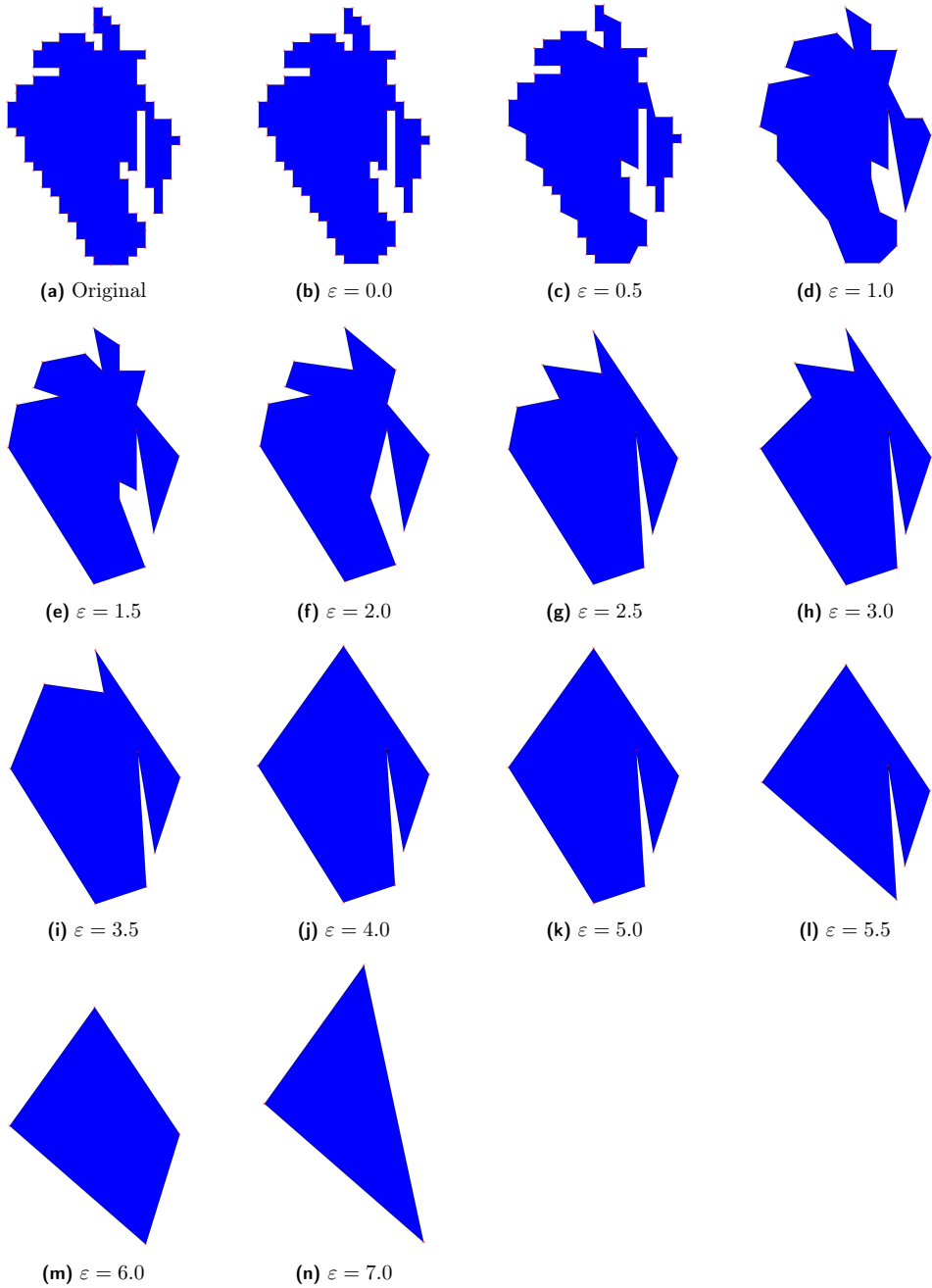
Zur rechnerischen Komplexität des Douglas-Peucker-Algorithmus sei bemerkt, dass die Laufzeit für die Reduktion des Polygons  $P$  mit  $n$  Knoten auf ein Polygon  $P'$  mit  $p$  Kanten im schlechtesten Fall  $\mathcal{O}(np)$  beträgt. Begründet wird dies durch die Tatsache, dass der Abstand jedes Knotens des Eingangspolygons zu jeder Kante des resultierenden Polygons berechnet werden muss. Die Berechnung des Abstands zweier Punkte im euklidischen Raum ist dabei eine Operation mit konstanter Rechenzeit. Eine Verbesserung der Laufzeit auf  $\mathcal{O}(n \log n)$  unter Ausnutzung der Eigenschaften der konvexen Hülle des Polygonzugs wurde in Hershberger und Snoeyink (1992) präsentiert.

In einer praxisnahen Anwendung wurde auf einem PC im Programm QGIS (Quantum GIS Development Team 2009) ein Datensatz mit 66 Polygonen in Echtzeit von 2343 Knoten auf 526 Knoten reduziert.

Ein Datensatz mit 118025 Polygonen wurde in 50s von rund 8.000.000 auf 2.000.000 Knoten reduziert. Wie man sieht ist auch die Verarbeitung großer Datenmengen mit relativ geringem Aufwand an Rechenzeit möglich.

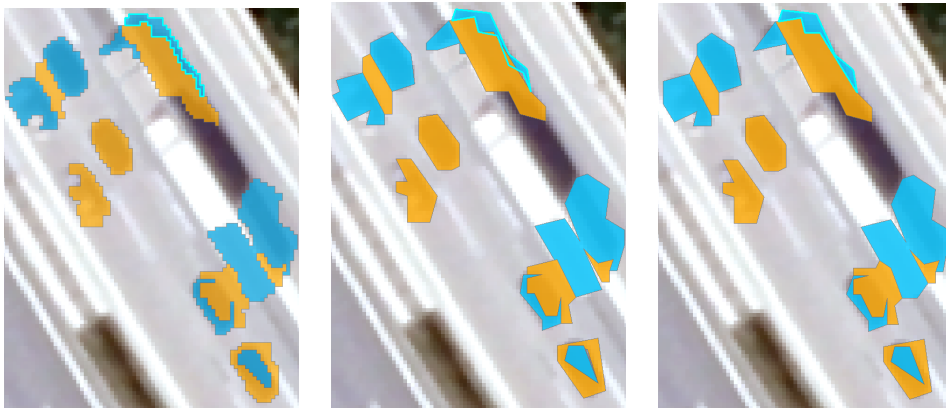
Da der Algorithmus Listen von Punkten verarbeitet, gelangt er bei geschlossenen Polygonzügen zu unterschiedlichen Ergebnissen je nach Auswahl des Start- und Endpunkts. Um dieses Problem zu beheben, kann man prinzipiell den Algorithmus für jeden Startpunkt einmal ausführen und anschließend das Ergebnis wählen, das dem Ursprungspolygon möglichst ähnlich ist. Eine Möglichkeit, die Ähnlichkeit zu quantifizieren, ist die in Abschnitt 2.8.2 vorgestellte Hausdorff-Metrik. Dieser Ansatz resultiert jedoch in einer Vervielfachung der Rechenzeit. Ein Ansatz, der diesen Aufwand mit Methoden aus dem Bereich der dynamischen Programmierung reduziert, wird in Kolesnikov und Fränti (2007) vorgeschlagen. Alternativ kann auch ein weniger kompliziertes Verfahren aus Ramer (1972) verwendet werden. Darin wird ein heuristischer Ansatz vorgeschlagen, der den obersten am weitesten links liegenden Punkt  $P_{UL}$  und den untersten am weitesten rechts liegenden Punkt  $P_{LR}$  als Startpunkte wählt, da diese vom Algorithmus wahrscheinlich sowieso erhalten werden würden. Ausgehend von diesen Startwerten wird das Verfahren unabhängig voneinander auf die Polygonzüge  $(P_{LR}, \dots, P_{UL})$  und  $(P_{UL}, \dots, P_{LR})$  angewandt.

Bei der klassischen Implementierung des Douglas-Peucker-Algorithmus kann zudem die unerwünschte Situation auftreten, dass sich Kanten des generalisierten Polygons gegenseitig schneiden. Für ein Beispiel sei auf Wu und Marquez (2003) verwiesen. Um dieses Verhalten des Algorithmus auszuschließen, werden in Saalfeld (1999) und Wu und Marquez (2003) Varianten des Verfahrens vorgeschlagen. Diese Ansätze nutzen die Tatsache, dass beim Vereinfachen eines Teilpolygonzugs nur Überschneidungen entstehen können, wenn sich die konvexe Hülle mit der konvexen Hülle eines anderen Teilpolygonzugs schneidet. Im Fall eines Überschneidens der konvexen Hüllen kann dann z.B. der Parameter  $\varepsilon$  für den entsprechenden Teilpolygonzug reduziert werden.



**Abb. 2.41.:** Beispielausgabe des Douglas-Peucker-Algorithmus für verschiedene Werte des Parameters  $\epsilon$ .

Bei der Verarbeitung von mehr als einem Polygon, kann es wichtig sein, die Topologie der Polygone untereinander zu erhalten. Werden z.B. zwei aneinander grenzende Polygone unabhängig voneinander vom Douglas-Peucker-Algorithmus vereinfacht, so kann nicht garantiert werden, dass zwischen den resultierenden Polygonen nicht eine Lücke entsteht. Das Problem lässt sich jedoch lösen, indem man die Polygone nicht nur als Liste von Punkten, sondern als Liste von Referenzen auf Begrenzungslinien repräsentiert. Polygone, die aneinander grenzen, referenzieren dann dieselbe Begrenzungslinie. Werden nun die Begrenzungslinien vom Algorithmus vereinfacht, so wirkt sich das auf alle Polygone aus, die auf diese Linie verweisen, wodurch die gemeinsame Grenze der Polygone erhalten bleibt (siehe Hahmann 2006). Um die Erzeugung von Überschneidungen von benachbarten Polygonen zu vermeiden, können erneut die erwähnten Techniken aus Saalfeld (1999) und Wu und Marquez (2003) verwendet werden. In der ArcGIS-Funktion „*Simplify Polygon*“ wird mit dem Parameter „*Handling Topological Errors*“ dem Nutzer eine Möglichkeit gegeben, eine Überprüfung der topologischen Korrektheit durchführen zu lassen. Bei Auswahl der Methode `RESOLVE_ERRORS` werden Topologiekonflikte gelöst, indem iterativ die Vereinfachung rückgängig gemacht, der Parameter  $\varepsilon$  halbiert und diese Schritte solange wiederholt werden, bis keine Konflikte mehr auftreten. Ein Beispiel dazu ist in Abbildung 2.42 dargestellt.



(a) Originalausgabe der Änderungsklassifizierung.

(b) Vereinfachung der Originalausgabe mittels Douglas-Peucker-Algorithmus ( $\varepsilon = 1.5$ ).

(c) Vereinfachung der Originalausgabe durch Douglas-Peucker-Algorithmus mit Auflösung von Topologiekonflikten ( $\varepsilon = 1.5$ ).

**Abb. 2.42.:** Beispielanwendung des Douglas-Peucker-Algorithmus mit Zwischenergebnissen.

Eine weitere Modifikation des Douglas-Peucker-Algorithmus wurde in Neumann und Selke (2001) vorgestellt. Dieser Ansatz hat die Erhaltung rechter Winkel im Ausgangspolygon zum Ziel und ist zur korrekten Extraktion von rechteckigen Strukturen besser geeignet, als der ursprüngliche Algorithmus. Das Verfahren beginnt mit der Suche nach den vier längsten Kanten des Polygons. Abschnitte zwischen den Kanten werden mit einem modifiziertem Douglas-Peucker-Algorithmus aufgerufen, der zuerst nach Scheitelpunkten rechter Winkel sucht und diese sowie deren benachbarte Punkte erhält. Erst anschließend wird nach Punkten mit dem größten Abstand von der Sekante gesucht.

Zudem existiert eine Variante des Douglas-Peucker-Algorithmus (siehe Ballard und Brown 1982, S. 234), bei der nicht rekursiv, sondern iterativ in jedem Durchlauf alle verbliebenen Punkte durchlaufen werden und unter diesen Punkten jener mit dem größten Abstand zur aktuellen Linie als Teilungspunkt ausgewählt wird. Wird kein Punkt mehr gefunden, der einen

Abstand größer als  $\varepsilon$  zum so konstruierten Polygonzug aufweist, stoppt der Algorithmus. Abschließend sei für eine weiterführende Zusammenfassung über Verfahren zur Polygonvereinfachung auf Heckbert und Garland (1997) und Abam u. a. (2010) verwiesen. Auch existieren in *eCognition* sogenannte Formfilter innerhalb des Algorithmus **pixel based shape processing filters**. Die genaue Funktionsweise wird jedoch im Handbuch zu *eCognition* (siehe Definiens Imaging, Munich 2009) nicht näher ausgeführt.

### 2.7.3. Zusammenfassung

Der vorangegangene Abschnitt hat aufgezeigt, dass im Nachgang der eigentlichen Änderungsdetektion einige abschließende Korrekturschritte angewandt werden sollten, damit das Analyseergebnis in geeigneter Form weiterverwendet werden kann. Der Fokus lag hierbei hauptsächlich auf der Reduktion der Änderungspolygone, die dadurch eine simplere Kontur erhalten, welche trotzdem die wesentliche Information repräsentiert.

Im Anschluss der Nachverarbeitung können die Daten in ein GIS importiert werden. Dadurch können weiterführende Fragestellungen etwa in Form dynamischer Visualisierungen beantwortet werden. Da die GIS-Verarbeitung der Daten aus der Änderungsdetektion nicht Bestandteil dieser Arbeit ist, sei dafür auf Daneke (2008) verwiesen.

## 2.8. Genauigkeitsuntersuchung

### 2.8.1. Einleitung

Zur Einschätzung des Verfahrens der objekt-basierten Änderungsdetektion kann eine Beurteilung der Ergebnisgüte hilfreich sein. Eine Aufteilung der Untersuchung in die Bewertung von Positions- und Formfehlern sowie thematischen Ungenauigkeiten, wie sie in Schiewe u. a. (2001) für GEOBIA allgemein vorgeschlagen wird, erscheint sinnvoll. Jedoch wäre ein solcher Prozess laut Stow (2010) für die objekt-basierte Änderungsdetektion aufgrund der zeitlichen als auch objekt-orientierten Art der Ergebnisse sehr anspruchsvoll.

Zu den prinzipiellen Schwierigkeiten der Gütebetrachtung kommt im Fall der Bewertung der Segmentierungsqualität hinzu, dass es keinen einheitlichen und allgemein in der Forschung akzeptierten Rahmen für eine solche Analyse gibt. In Unterabschnitt 2.8.2 wird daher ein Überblick der dafür verfügbaren Bewertungsmethoden gegeben.

Über die separate Betrachtung der Fehlerarten hinaus existiert jedoch auch die Möglichkeit beide Fehler zusammen gemäß der Methode *Point in Polygon* zu betrachten, die unter anderem in Conchedda u. a. (2008) und Zhou u. a. (2008) zum Einsatz kommt. Die Idee dahinter ist die zufällige und über dem gesamten Bildbereich gleichverteilte Auswahl einzelner Punkte, für welche der Klassifikationsfehler des zugehörigen Objekts begutachtet wird. Mit der Erläuterung dieses Verfahrens beschäftigt sich der Unterabschnitt 2.8.3.

### 2.8.2. Güte der Segmentierung

Da die Segmentierung der Bilddaten einen wesentlichen Schritt bei der objekt-basierten Änderungsdetektion darstellt, ist folglich auch die Betrachtung der Qualität des Ergebnisses dieses Schrittes wichtig. Insbesondere bei der Auswahl der Parameter oder sogar des Algorithmus selbst kann eine solche Analyse hilfreich sein.

Zunächst stellt sich die Frage, was Segmentierungsgüte eigentlich bedeutet. Wählt man einen qualitativen Ansatz, muss man in der Folge mit subjektiven Einschätzungen des Anwenders umgehen und erhält keine abschließende Lösung. Daher sind Verfahren vorzuziehen, welche die Segmentierungsgüte quantitativ bewerten. Jedoch führt dies abhängig vom verwendeten

Kriterium dazu, dass mutmaßlich jene Segmentierungsalgorithmen, die das Bewertungskriterium als Grundlage für eine Optimierung einsetzen, am besten abschneiden.

Trotz des genannten Nachteils eignen sich quantitative Verfahren, um die Güte der Segmentierung zu evaluieren. In der Literatur wurde dazu eine Vielzahl an Methoden vorgeschlagen. Prinzipiell lassen sich diese Verfahren in zwei Gruppen einteilen.

Die erste Gruppe umfasst sogenannte unüberwachte Ansätze zur Evaluierung der Güte. Bei diesen Methoden werden allein unter Verwendung der Bilddaten sowie des Ergebnisses der Segmentierung Indizes berechnet, welche versuchen, die Qualität der Segmentierung zu quantifizieren. Als Beispiele für diese Art von Verfahren ist einerseits der entropie-basierte Ansatz in Zhang u. a. (2004) zu nennen, der im Verlauf dieses Abschnitts im Detail beschrieben wird. Andererseits sei der *Near Set Index* genannt, ein Verfahren zur Abschätzung der Helligkeitsvariabilität innerhalb eines Segments, welches in Henry und Peters (2008) beschrieben wird. Eine umfassende Analyse der bestehenden, unüberwachten Verfahren zur Bewertung der Segmentierungsgüte findet sich in Zhang u. a. (2008).

Die zweite Gruppe von Evaluierungsverfahren sind überwachte Methoden. Prinzipiell wird dabei die automatisch generierte Segmentation insgesamt oder in Teilen mit vom Benutzer spezifizierten Beispielsegmenten verglichen. Eine gute automatisch generierte Segmentation zeichnet sich bei diesem Ansatz durch eine hohe Übereinstimmung mit den Referenzsegmenten aus. Die einzelnen Verfahren dieser Gruppe unterscheiden sich meist in der Methode zur Quantifizierung der Unterschiede. Beispielsweise verwendet Feitosa u. a. (2006) eine sogenannte Fitnessfunktion, welche die symmetrische Differenz zwischen Referenzsegment und dem automatisch generierten Segment mit der größten Überlappung zum Referenzsegment berechnet. Je größer diese Fläche relativ zur Fläche des Referenzsegments ist, umso besser wird die Güte des Verfahrens eingeschätzt. Eine Erweiterung dieser Idee auf mehrere Maßzahlen, welche einen Gradmesser für Über- bzw. Untersegmentierung liefern sollen, findet sich in Marpu u. a. (2010). Darüber hinaus werden Distanzmaße wie der *Normalized Probabilistic Rand Index* (siehe Unnikrishnan u. a. 2007) und die Hausdorff-Metrik (siehe Beauchemin u. a. 1998) zum Vergleich von Segmentationen eingesetzt.

Wichtig für die Art der Gütebetrachtung ist neben den Bewertungsverfahren insbesondere für den Vergleich von Segmentierungsverfahren eine einheitliche Bilddatenbasis, welche durch möglichst vielen Entwicklern von Segmentierungsalgorithmen herangezogen wird. Eine Datenbank von Texturmustern findet sich dabei in Haindl und Mikes (2008). Photographien allgemeiner Art werden hingegen in Martin u. a. (2001) bereitgestellt. Spezielle Datenbanken für Fernerkundungsaufnahmen existieren hingegen nach Kenntnisstand des Autors nicht.

Nichtsdestoweniger existieren in der Literatur auch für den Anwendungsbereich der Fernerkundung vergleichende Analysen von Segmentierungsverfahren. Insbesondere die Studien in Meinel und Neubert (2004), Neubert u. a. (2008) und Marpu u. a. (2010) betrachten das Problem der Segmentierungsgüte mithilfe verschiedener Bewertungsansätze und einer einheitlichen Datenbasis bestehend aus einzelnen Fernerkundungsaufnahmen. Dabei kommen in diesen Arbeiten sowohl überwachte als auch unüberwachte Evaluierungsverfahren zum Einsatz. In diesen Arbeiten wird zudem gezeigt, dass der in Abschnitt 2.3.3 präsentierte *Multiresolution-Segmentation*-Algorithmus gute Ergebnisse für die Segmentierung von Fernerkundungsdaten liefert. Weitere Studien zur Segmentierungsqualität im Anwendungsgebiet der Fernerkundung finden sich in Van Coillie u. a. (2008b), Carleer u. a. (2005), Karantzalos und Argialas (2003) und Clinton u. a. (2010).

Neben reinen überwachten oder unüberwachten Verfahren kommen zur Segmentierungsbewertung im Rahmen der Fernerkundung auch Mischformen der beiden Ansätze vor. Beispielsweise verwendet Kim und Madden (2006) die lokale Varianz als Gütekriterium und diese der Klassifizierungsgenauigkeit gegenüber. Benz u. a. (2004) hingegen schlägt eine Mischbewertung für Stabilität und überwachter Güte der Segmentierung vor. Als Stabilitätsmaß dient dabei das Verhältnis der starken zu den schwachen Außenkanten eines Segments. Für die Güte der au-

tomatischen Segmentierung eines Referenzsegments wird die Über- und Untersegmentierung mit Kennzahlen ähnlich wie in Marpu u. a. (2010) abgeschätzt.

Methoden zur Untersuchung der Segmentierungsgüte kommen außer für die Auswahl eines Segmentierungsverfahrens auch bei der Wahl der Parameter der Algorithmen zum Einsatz. Für diese Arbeit stellt sich insbesondere die Frage nach der Wahl des Parameters  $T$  für den MRS-Algorithmus (siehe Abschnitt 2.3.3).

Zur Lösung dieses Problems bietet sich einerseits an, auf Grundlage von verschiedenen Konfigurationen, den Parameter qualitativ vom Nutzer bestimmen zu lassen. Dieser Ansatz wird in vielen Arbeiten zur objekt-basierten Bildanalyse gewählt, kann jedoch die Ergebnisqualität nicht garantieren und leidet an subjektiven Verzerrungen. Deshalb kommen bei vielen Segmentierungsverfahren Referenzsegmente zum Einsatz. Beispielhaft sind dafür der Segmentierungsalgorithmus in *ENVI Fx* (siehe Solutions 2008), das *SIOX*-Verfahren (siehe Friedland u. a. 2005) und der Algorithmus aus Poggi u. a. (2005) zu nennen. Seit der *eCognition*-Version 8.64 kann eine solche überwachte Methode auch in der Software selbst mithilfe des Map-Konzeptes realisiert werden. Trotzdem bleibt bei auch diesem Ansatz ein Rest an Subjektivität durch die Notwendigkeit der Auswahl von Referenzsegmenten durch den Anwender. Schließlich können auch unüberwachte Evaluierungsverfahren zur Bestimmung des Parameters  $T$  genutzt werden wie in Draguta u. a. (2010) anhand der Varianzänderung innerhalb der Segmente gezeigt wird.

Im Rahmen dieser Arbeit kommt ein unüberwachtes Verfahren auf Basis der Entropie für die Evaluierung der Segmentierungsgüte und die Bestimmung des Segmentierungsparameters  $T$  zum Einsatz, das dem Ansatz in Zhang u. a. (2004) entlehnt ist. Die Motivation für den Einsatz des Verfahrens liegt in der einfachen Implementierbarkeit des Verfahrens sowie seiner statistischen Fundierung. Außerdem werden keine Referenzsegmente benötigt, was die automatisierte Anwendung erleichtert.

Zhangs Verfahren zur Evaluierung der Güte einer Segmentation nutzt die informationstheoretische Entropie als Ausgangspunkt. Für ein beliebiges Segment  $s$  einer Segmentation  $S$  mit der relativen Häufigkeit  $h$  der Merkmalsausprägungen  $m_X \in M_X(s)$  innerhalb eines Segments kann die Entropie  $H$  ermittelt werden durch

$$H(s) = - \sum_{m_X \in M_X(s)} h_s(m_X) \log h_s(m_X). \quad (2.136)$$

Das Merkmal  $X$  bezieht sich hierbei auf einen Bildkanal bzw. einen Pixelwert, wobei in Zhang u. a. (2004) die Verwendung des quadratischen Mittels über alle Kanäle vorgeschlagen wird.  $H(s)$  gibt dann an, welche Bitlänge zur Kodierung der Information im Segment  $s$  mindestens nötig ist oder alternativ welcher Informationsgehalt in  $s$  enthalten ist.

Auf dieser Grundlage kann für das Gesamtbild die Segment-Entropie als gewichtete, mittlere Entropie werden durch

$$H_r(S) = \sum_{s \in S} \frac{n_s}{n} H(s). \quad (2.137)$$

$H_r$  gibt somit den größenbereinigten, durchschnittlichen Informationsgehalt eines Bildbereichs bei Verwendung der Segmentation  $S$  wieder. Für eine gute Segmentation sollte dieser Wert möglichst niedrig sein, da möglichst homogene Segmente gesucht werden. Jedoch wird bei sinkender Anzahl der Segmente eines Bildes, der Wert für  $H_r$  immer größer.

Da eine reine Minimierung von  $H_r$  immer in einer Segmentation resultieren würde, bei der jedes Pixel ein einzelnes Segment darstellt, wird eine zweite Größe als Korrekturterm  $H_l$  definiert durch

$$H_l(S) = - \sum_{s \in S} \frac{n_s}{n} \log \frac{n_s}{n}. \quad (2.138)$$

Diese Größe entspricht der Entropie des Merkmals „relative Objektgröße“ in der Segmentation  $S$  und spiegelt somit den Informationsgehalt der Segmentation  $S$  selbst wider. Je größer die

Objektanzahl einer Segmentation ist, umso größer wird auch der Wert von  $H_r$ . Aggregiert man die beiden Indikatoren  $H_r$  und  $H_l$  in einem gewichteten Mittel, so erhält man

$$E(S) = w_l H_l(S) + (1 - w_l) H_r(S). \quad (2.139)$$

Durch das Gewicht  $w_l \in [0, 1]$  kann gesteuert werden, welcher der beiden Terme betont werden soll.

Eine gute Segmentation des Bildes zeichnet sich einerseits durch möglichst große Objekte aus, was in einem geringen Wert für  $H_l$  resultiert und andererseits sollten die einzelnen Objekte möglichst homogen sein, was zu einem geringen Wert für  $H_r$  führt. Insgesamt ist eine Segmentation  $S$  als gut anzusehen, wenn der Wert für  $E(S)$  möglichst niedrig ist.

Die beschriebene Methode bietet keinen absoluten Maßstab für die Güte einer Segmentation. Es ist jedoch möglich mit diesem Ansatz Segmentationen, die mithilfe verschiedener Verfahren oder Parameterwerte erzeugt wurden, miteinander zu vergleichen.

Für die Realisierung im Rahmen dieser Arbeit kann das Entropie-Plugin für *eCognition*, das im Abschnitt 2.4.3 präsentiert wurde, eingesetzt werden, um den Wert für die Layout-Entropie  $H_l$  zu bestimmen. Für die Berechnung der Segment-Entropie  $H_r$  ist jedoch ein weiteres Plugin nötig, das auf der beiliegenden CD im Verzeichnis **LayerEntropy** zu finden ist.

Zur Wahl des Segmentierungsparameters  $T$  kann das Verfahren eingesetzt werden, indem über den Wertebereich des Parameters iteriert wird und dabei für jeden Wert von  $T$  der Güteindex  $E$  berechnet wird. Daraus schätzt man eine Funktion für  $E(T)$ . Jene Stellen im Definitionsbereich der Funktion mit lokalen Minima können laut Zhang u. a. (2004) als geeignete Parameter verwendet werden, da diese einer Segmentation den Vorzug in einem bestimmten Parameterbereich geben. Auf diese Weise ist es zwar nicht möglich einen eindeutig besten Parameterwert zu bestimmen. Es können jedoch Parameter für Maßstabsbereiche ermittelt werden.

Neben der Implementierung des Verfahrens wurden erste Versuche mit dem Verfahren im Rahmen der Entwicklung des Plugins durchgeführt, die im Folgenden erläutert werden. Die dafür verwendeten Bilddaten sind identisch zu den in Abschnitt 2.3.4 eingesetzten Aufnahmen.

Ein erster Versuch untersucht für das Problem der Einzelbildsegmentierung vergleichsweise die Ergebnisse einer Schachbrettsegmentierung, des MRS-Algorithmus sowie der im Rahmen dieser Arbeit reimplementierten Version des MRS-Algorithmus. Die Parameter wurden so gewählt, dass die Objektanzahl für alle drei Segmentationen konstant ist, damit eine Vergleichbarkeit gewährleistet werden kann.

Die Ergebnisse sind in Tabelle 2.3 dargestellt. Sie zeigen deutlich, dass die Güte der Schachbrettsegmentierung unter der des MRS-Algorithmus liegt. Außerdem ist die Güte des reimplementierten MRS-Algorithmus auf gleicher Höhe mit der Originalversion auch wenn die Resultate für  $E$  nicht exakt übereinstimmen.

	$n$	$T$	$H_l$	$H_r$	$E$				
$w_l$					0,1	0,3	0,5	0,7	0,9
Schachbrettsegmentierung	600	30	6.39	4.55	4.73	5.10	5.47	5.84	6.21
MRS4CD-Algorithmus	601	50.1	5.55	4.14	4.28	4.56	4.84	5.12	5.40
MRS-Algorithmus	601	51.4	5.51	4.15	4.28	4.55	4.83	5.10	5.37

**Tab. 2.3.:** Vergleich der Güte verschiedener Segmentierungsverfahren.

Ein zweiter Versuch untersucht, ob der MRS4CD-Algorithmus neben der qualitativen Betrachtung auch beim Güteindikator  $E$  gute Ergebnisse zeigt. Dazu werden die Ergebnisse der drei Ansätze zur direkten Nutzung des MRS-Algorithmus für die Änderungsdetektion<sup>15</sup>

<sup>15</sup>Die erste Variante segmentiert das Bild auf Grundlage der Bilddaten des ersten Aufnahmezeitpunkts. Variante zwei nutzt die Daten beider Zeitpunkte gleichzeitig. Die dritte Variante segmentiert die Daten beider

mit den Resultaten des MRS4CD-Algorithmus für beide Aufnahmezeitpunkte verglichen. Zur Kontrolle wird auch die Güteabschätzung für eine Schachbrettsegmentierung herangezogen. Die zusammengefassten Ergebnisse werden in Abbildung 2.4 dargestellt. Auch wenn diese Resultate nicht repräsentativ sind, geben sie doch einen Einblick darin, dass der MRS4CD-Algorithmus Ergebnisse produziert, die einer Evaluierung standhalten. Insbesondere werden bessere Ergebnisse erzielt als wenn nur ein Bild oder beide Bilder gleichzeitig als Eingabe des MRS-Algorithmus verwendet werden.

	Bild	$n$	$T$	$H_l$	$H_r$	$E$				
$w_l$						0,1	0,3	0,5	0,7	0,9
Schachbrettsegmentierung	1	600	30	6.39	4.55	4.73	5.10	5.47	5.84	6.21
MRS-Variante 1	1	601	51.4	5.51	4.15	4.28	4.55	4.83	5.10	5.37
MRS-Variante 2	1	612	47.5	5.63	4.15	4.30	4.59	4.89	5.19	5.48
MRS-Variante 3	1	601	51.4	5.51	4.15	4.28	4.55	4.83	5.10	5.37
MRS4CD	1	601	50.1	5.55	4.14	4.28	4.56	4.84	5.12	5.40
Schachbrettsegmentierung	2	600	30	6.39	4.55	4.73	5.10	5.47	5.84	6.21
MRS-Variante 1	2	601	51.4	5.51	4.16	4.30	4.57	4.84	5.10	5.37
MRS-Variante 2	2	612	47.5	5.63	4.14	4.29	4.59	4.89	5.18	5.48
MRS-Variante 3	2	612	47	5.61	4.13	4.28	4.58	4.87	5.17	5.46
MRS4CD	2	611	47.42	5.56	4.13	4.28	4.56	4.84	5.13	5.41

Tab. 2.4.: Vergleich der Güte verschiedener bitemporaler Segmentierungsverfahren.

Da die hier vorgestellte Methode auch zur Bestimmung des Parameters  $T$  eingesetzt werden kann, soll auch für diesen Anwendungsfall im Folgenden ein Beispiel gezeigt werden. Dazu werden die Bilddaten des ersten Aufnahmezeitpunktes mit dem MRS-Algorithmus segmentiert. Der Parameter  $T$  variiert dabei zwischen 0 und 100. Pro Iterationsschritt wird  $T$  um den Wert 0.25 erhöht. Abbildung 2.43 zeigt den Verlauf der Funktion  $E$  bei Verwendung des Wichtungsparameters  $w_l = 0.5$ .

In dieser Darstellung sind mit bloßem Auge zwar keine lokalen Minima erkennbar, jedoch lassen sich anhand der Daten vier Minima bestimmen, die für alle Werte von  $w_l$  stabil sind. In Abbildung 2.44 sind diese Segmentationen dargestellt.

Im Rahmen dieser Arbeit wird eine Analyse der Segmentierungsgüte für eine realitätsnahe Fallstudie mithilfe der hier vorgestellten Methode in Abschnitt 3.3.7 durchgeführt.

Abschließend sei bemerkt, dass die Evaluierung von Segmentationen eine große Rolle für das Gesamtergebnis spielt. Daher sollten neue Segmentierungsalgorithmen immer eine quantitative Gütebetrachtung durchführen. Dazu ist es jedoch nötig, eine gemeinsame Datenbasis zu schaffen. Um Verzerrungen der Ergebnisse durch die nutzer-basierte Definition von Referenzsegmenten zu vermeiden, können dazu simulierte Daten genutzt werden.

### 2.8.3. Güte der Änderungsklassifikation

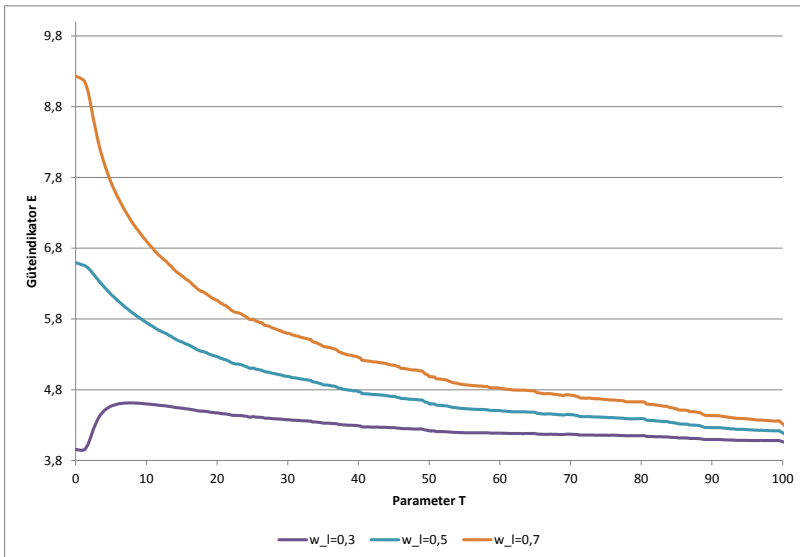
Im Gegensatz zur Segmentierung ist die Bewertung der Güte einer Klassifikation de facto standardisiert. Dazu kommen Referenzobjekte für jede Klasse zum Einsatz, die durch den Nutzer definiert oder einer anderen Quelle etwa einer bestehenden Karte entnommen werden. In vielen Arbeiten wird darauf aufbauend die Genauigkeit einer Klassifikation auf Grundlage der sogenannten Konfusionsmatrix betrachtet.

In einer Konfusionsmatrix werden für jede Spalte die Referenzobjekte einer Klasse jener Zeile zugeordnet, die dem Ergebnis der automatischen Klassifizierung entspricht. Eine Eintragung

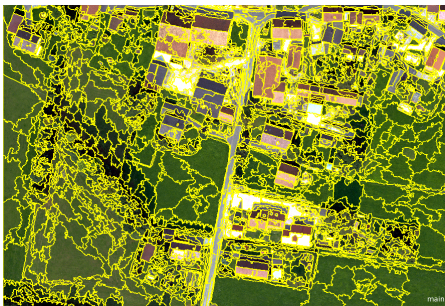
---

Aufnahmezeitpunkte unabhängig. Details zu den drei Varianten der Nutzung des MRS-Algorithmus im Rahmen der objekt-basierten Änderungsdetektion sei auf Abschnitt 2.3.4 verwiesen.

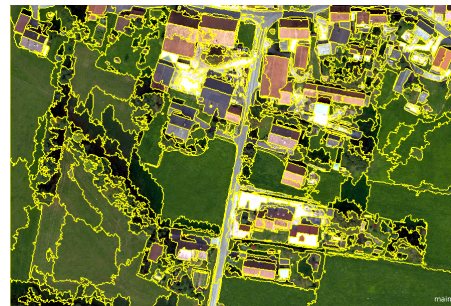




**Abb. 2.43.:** Segmentierungsgüte  $E$  in Abhängigkeit vom Parameter  $T$ .



(a)  $T = 24.75$  mit 1941 Segmenten.



(b)  $T = 55.5$  mit 513 Segmenten.



(c)  $T = 69$  mit 375 Segmenten.



(d)  $T = 78.75$  mit 308 Segmenten.

**Abb. 2.44.:** Vier Segmentationen mit lokalen Minima des Güteindicators  $E$ .

in der  $i$ -ten Zeile und  $j$ -ten Spalte repräsentiert somit die Anzahl der Referenzsegmente der Klasse  $j$ , welche durch den Klassifikator der Klasse  $i$  zugeordnet wurden. Die Summe einer Spalte der Matrix ist somit die Gesamtanzahl der Referenzsegmente für eine bestimmte Klasse. Die Zeilensumme der  $i$ -ten Matrixzeile hingegen gibt die Anzahl aller Referenzobjekte wieder, welche durch die automatische Klassifizierung der Klasse  $i$  zugeordnet worden sind. Die Gesamtanzahl der Referenzsegmente sei durch  $N$  gegeben.

Aus der so konstruierten Konfusionsmatrix  $C = (c_{i,j})_{i,j=1,\dots,n}$  lassen sich intuitiv verständliche Maßzahlen für die Güte einer Klassifikation ableiten. Dazu seien durch  $c_{i,*}$  die Zeilensumme der  $i$ -ten Zeile und durch  $c_{*,j}$  die Spaltensumme der  $j$ -ten Spalte gegeben.

Auf dieser Grundlage kann die Gesamtgenauigkeit über die Einträge der Hauptdiagonalelemente im Verhältnis zur Gesamtsumme abgeschätzt werden, d.h.

$$OA = \frac{\sum_{i=1}^n c_{i,i}}{N}. \quad (2.140)$$

Für eine beliebige Klasse  $j$  lässt sich die sogenannte *Producer's Accuracy* also die „Genauigkeit aus Sicht des Produzenten“ definieren als der Anteil der Referenzsegmente für  $j$ , welche auch als  $j$  klassifiziert werden. Dies entspricht der Genauigkeit bezogen auf die Referenzobjekte. Berechnet werden kann diese Maßzahl spaltenweise durch das Verhältnis des Hauptdiagonalelements zur jeweiligen Spaltensumme, d.h.

$$PA_j = \frac{c_{j,j}}{c_{*,j}}. \quad (2.141)$$

Analog kann auch die „Genauigkeit aus Sicht des Nutzers“, die sogenannte *User's Accuracy*, für eine Klasse  $i$  bestimmt werden durch

$$UA_i = \frac{c_{i,i}}{c_{i,*}}. \quad (2.142)$$

Diese Maßzahl gibt an, wie groß der korrekt klassifizierte Anteil der Kartenobjekte ist, die der Klasse  $i$  zugeordnet wurden. Das entspricht der Genauigkeit bezüglich der Karte.

Ein weiteres Kriterium für die Güte einer Klassifikation, das aus der Konfusionsmatrix abgeleitet werden kann, ist der Kappa-Index. Definiert wird diese Maßzahl als

$$\hat{K} = \frac{N \sum_{i=1}^n c_{i,i} - \sum_{i=1}^n (c_{i,*} c_{*,i})}{N^2 - \sum_{i=1}^n (c_{i,*} c_{*,i})}. \quad (2.143)$$

Der Kappa-Index kann interpretiert werden als Übereinstimmung zwischen Referenzklassifikation und automatisch generierter Klassifikation, die über eine zu erwartende zufällige Übereinstimmung hinausgeht.

Ein Beispiel für eine Konfusionsmatrix einschließlich einiger Maßzahlen ist in Abbildung 2.45 dargestellt.

Für detailliertere Erläuterungen zur Konfusionsmatrix und den daraus abgeleiteten Güteindikatoren sei auf Lillesand u. a. (2008, S. 585-592) verwiesen.

Generell ist festzustellen, dass sowohl für das Training als auch für die Gütebetrachtung von Klassifikatoren Beispielobjekte benötigt werden. Um jedoch verlässliche Aussagen über die Güte treffen zu können, sind die Testobjekte strikt von den Trainingsobjekten zu trennen.

In Bezug auf die objekt-basierte Bildanalyse ist festzustellen, dass in der Literatur als Grundlage für die Untersuchung der Klassifizierungsgenauigkeit sowohl Pixel (siehe Im u. a. 2008; Conchedda u. a. 2008) als auch Objekte (siehe Desclee u. a. 2006; Stow u. a. 2008) verwendet werden. Laut Chen u. a. (2012) unterschätzen pixel-basierte Gütebetrachtungen die Genauigkeit objekt-basierter Klassifikationen. Jedoch sind gemäß derselben Quelle objekt-basierte Genauigkeitsuntersuchungen vornehmlich geeignet für Analysen, in denen es nur eine Art von

	Referenz				User's Accuracy
		Klasse 1	Klasse 2	Klasse 3	
Klassifikator	Klasse 1	5	2	0	$5/(5+2+0) = 71\%$
	Klasse 2	3	3	2	$3/(3+3+2) = 37,5\%$
	Klasse 3	0	1	11	$11/(0+1+11) = 91,7\%$
	Producer's Accuracy	$5/(5+3+0) = 62,5\%$	$3/(2+3+1) = 50\%$	$11/(0+2+11) = 84,6\%$	$(5+3+11) / (5+2+0+3+3+2+0+1+11) = 70,4\%$

**Abb. 2.45.:** Beispiel für eine Konfusionsmatrix zur Evaluierung der Klassifizierungsgenauigkeit.

Objekten von Interesse gibt. Andernfalls werde die Analyse wesentlich komplizierter als im pixel-basierten Fall. Zudem trete für die objekt-basierte Änderungsdetektion das Problem eines nur teilweise von Änderungen betroffenen Objekts auf.

Marpu (2009) hingegen lehnt eine objekt-basierte Genauigkeitsuntersuchung unter Verwendung der automatisch Segmentation prinzipiell ab, da die Ergebnisse einer Analyse mit den ursprünglichen Messungen verglichen werden müssten. Diese Messungen sind jedoch durch Pixel und nicht durch Objekte gegeben. Motiviert durch dieses Argument wird auch im Rahmen der vorliegende Arbeit die Änderungsklassifizierung anhand der Pixel bewertet.

#### 2.8.4. Zusammenfassung

Der Prozess der objekt-basierten Änderungsdetektion kann durch die im vorangegangenen Methoden zur Gütebewertung begleitet werden. Die Segmentierungsgüte sollte dabei unabhängig von der Genauigkeit des Endergebnisses betrachtet werden. Dadurch ist gewährleistet, dass der Einfluss der Segmentierungsfehler auf das Gesamtergebnis reduziert wird.

Darüber hinaus ist jedoch die Genauigkeitsuntersuchung der Änderungsklassifikation besonders wichtig, um die Vergleichbarkeit zu anderen Studien herzustellen. Eine pixel-basierte Gütebewertung der Änderungsklassifizierung wird daher im Rahmen der Fallstudien in Kapitel 3 durchgeführt.

### 2.9. Zusammenfassung

In diesem Kapitel wurde eine umfassende Darstellung der Methoden für die objekt-basierte Änderungsdetektion gegeben. Das Rüstzeug für den Einsatz einer vollständig objekt-basierten Methode der Änderungsdetektion wurde hierfür präsentiert. Passende Methoden wurden im Rahmen des Kapitels beschrieben, motiviert und ausgewählt. Eine Implementierung der relevanten Verfahren findet sich auf der dieser Arbeit beiliegenden CD.

Darüber hinaus konnte die Funktionsweise des Prozesses bereits in diesem Kapitel anhand simulierter Änderungsdaten erläutert werden. Mit der Anwendung der Methoden auf realistische Fallbeispiele beschäftigt sich das folgende Kapitel 3.

## 3. Anwendung

### 3.1. Einleitung

Ziel dieses Kapitels ist die Überprüfung der Anwendbarkeit der im vorangegangenen Kapitel vorgestellten Methode zur objekt-basierten Änderungsdetektion auf reale Datensätze. Es soll dabei ein Vergleich der Methoden auf Grundlage der Anwendung erfolgen. Die vergleichenden Untersuchungen haben die Betrachtung verschiedener Aspekte zum Ziel. Erstens wird ein Vergleich der beiden Paradigmen, der objekt- sowie der pixel-basierten Bildanalyse, für die Änderungsdetektion, durchgeführt. Zweitens werden die unterschiedlichen Prinzipien der Segmentierung für die Änderungsdetektion der vollständig objekt-basierten Methode aus Kapitel 2 gegenübergestellt. Drittens wird eine Untersuchung der Algorithmen DCD und CDIO zur Objektentsprechung zwischen unterschiedlichen Segmentationen stattfinden. Schließlich werden viertens diverse Arten der Transformation des Merkmalsraumes wie PCA und MAD sowie deren Kombinationen in der praktischen Anwendung betrachtet.

Untersucht werden soll ebenfalls, welche Produkte den Analysten in seiner Aufgabe unterstützen. Das heißt, neben der eigentlichen Karte der Änderungen als Klassifikationsergebnis, werden auch Zwischenergebnisse wie die Änderungsintensität nach der Transformation des Merkmalsraumes als mögliche Hilfestellung in Erwägung gezogen.

Die Anwendungsfälle werden darüber hinaus von folgenden Fragen geleitet:

- Lassen sich durch den Einsatz von Form- und Textureigenschaften bessere Ergebnisse erzielen?
- Verbessert sich die Qualität der Segmentation im Vergleich zu den oben beschriebenen drei Möglichkeiten durch Integration des MRS4CD-Algorithmus?
- Wenn ja, kommt es durch die bessere Qualität der Segmentation zu besseren Ergebnissen bei der Änderungsdetektion?
- Wie können überwachte und unüberwachte Methoden zur Änderungsklassifizierung in der Praxis umgesetzt werden?

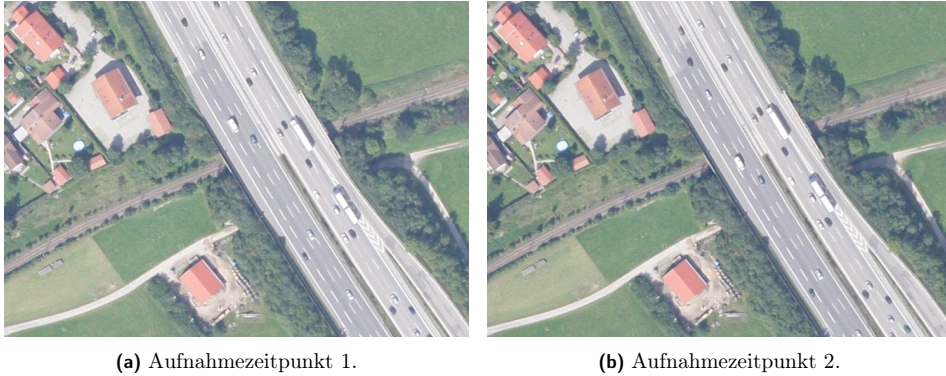
Um die genannten Aspekte und Fragestellungen zu untersuchen, werden im Folgenden zwei Fallstudien durchgeführt. Eine Diskussion der Ergebnisse findet im Anschluss an die jeweilige Studie statt. Das Kapitel schließt mit einer Zusammenfassung der gewonnenen Erkenntnisse.

### 3.2. Fallstudie „Autobahn“<sup>1</sup>

#### 3.2.1. Einleitung

Nach den im Rahmen des Abschnitts 2 verwendeten simulierten Daten soll die Funktionsweise der Methode zur objekt-basierten Änderungsdetektion anhand realer Daten gezeigt werden. Dazu kommen im Folgenden zwei Luftbilder einer Autobahn in Süddeutschland zum Einsatz, die mit einem zeitlichen Abstand von  $0.7s$  aufgenommen worden sind. Die verwendeten Originaldaten, die aus jeweils drei Kanälen bestehen, sind in Abbildung 3.1 dargestellt.

<sup>1</sup>Die Grundlage für diesen Abschnitt bildet Listner und Niemeyer (2011).



**Abb. 3.1.:** Für die Änderungsanalyse in Abschnitt 3.2 verwendete Bilddaten.

Durch den geringen Unterschied der Aufnahmezeitpunkte sind relevante Änderungen zwischen den Bildern nur im Bereich der Fahrbahn zu beobachten. Trotzdem können Sensorrauschen, die leicht unterschiedlichen Aufnahmewinkel sowie Unzulänglichkeiten der Koregistrierung der beiden Bilder zu Fehlern in der automatischen Änderungsdetektion führen.

In den folgenden Unterabschnitten werden die einzelnen Verarbeitungsschritte im Detail erläutert. Dabei ist das Ziel der Untersuchung, den Einfluss des Segmentierungsverfahrens auf die Ergebnisse der Änderungsdetektion zu bewerten. Hierzu werden drei Verarbeitungsvarianten betrachtet, die unterschiedliche Arten der Segmentierung sowie der Objektentsprechung beinhalten. In Variante A wird der MRS4CD-Algorithmus in Verbindung mit der Methode DCD zur Objektentsprechung benutzt. Variante B beinhaltet ebenfalls den MRS4CD-Algorithmus jedoch mit dem Verfahren CDIO zur Objektentsprechung. Bei der Variante C schließlich wird der MRS-Algorithmus auf Grundlage der pixel-basierten MAD-Variante ausgeführt, woraus nur eine Segmentation resultiert und somit keine Objektentsprechung zwischen mehreren Segmentationen benötigt wird.

### 3.2.2. Vorverarbeitung

Um Fehler in den Ergebnissen der Änderungsdetektion zu vermeiden, wird vor Beginn des eigentlichen Änderungsdetektionsprozesses eine Vorverarbeitung durchgeführt. Diese Vorverarbeitung umfasst erstens eine Medianfilterung (siehe Abschnitt 2.2) mit einem quadratischen Filterkern der Seitenlänge 3 px. Zweitens werden die Bilder radiometrisch mit der IR-MAD-Methode aus Canty u. a. (2004) korrigiert. Eine Koregistrierung der Daten muss im Rahmen dieser Arbeit nicht erfolgen, da die Bilder bereits geometrisch korrigiert vorliegen.

Die beschriebenen Vorverarbeitungsschritte sind für alle drei Verarbeitungsvarianten identisch.

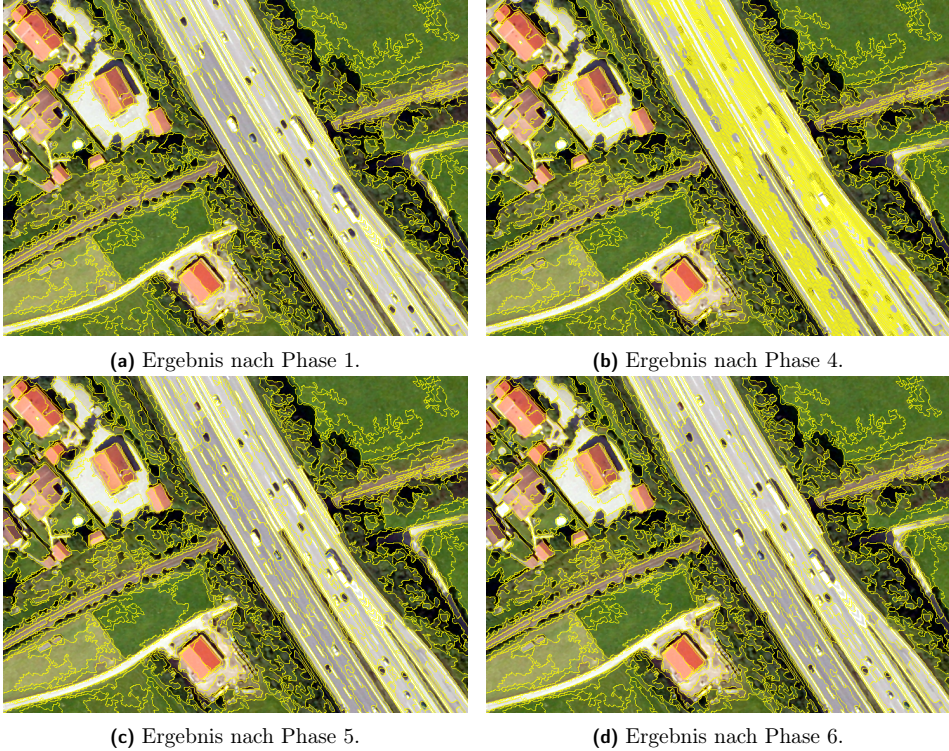
### 3.2.3. Segmentierung

Die Segmentierung der Bilder folgt als nächster Verarbeitungsschritt.

In den Varianten A und B wird der MRS4CD-Algorithmus benutzt. Dazu kommt der *Threshold Test* auf Basis der Bilddaten des zweiten Aufnahmezeitpunkts zum Einsatz. Inkonsistente Segmente werden mithilfe der Bereinigungsstrategie *Universal Removal Strategy* entfernt. Für die Phasen 1 und 5 des MRS4CD-Algorithmus (siehe Tabelle 2.1) kommen dazu die Parameter  $T = 50$  und  $w_s = 0$  zum Einsatz. In Phase 3 wird ein Heterogenitätsschwellenwert von  $T_t = 58$  benutzt. Für die abschließende Phase 6 wird ein Heterogenitätsschwellenwert von

$T = 25$  verwendet.

Das Ergebnis der Segmentierung für die Verarbeitungsvarianten A und B ist in Abbildung 3.2 dargestellt.



**Abb. 3.2.:** Segmentierung mithilfe des MRS4CD-Algorithmus für die Verarbeitungsvarianten A und B.

Für Verarbeitungsvariante C findet der MRS-Algorithmus mit den Parametern  $T = 3.5$ ,  $w_s = 0.01$  und  $w_{comp} = 0.5$  Anwendung. Die Datengrundlage bilden hier jedoch nicht die Bilddaten der beiden Aufnahmezeitpunkte, sondern die MAD-Variate einer pixel-basierten IR-MAD-Transformation.

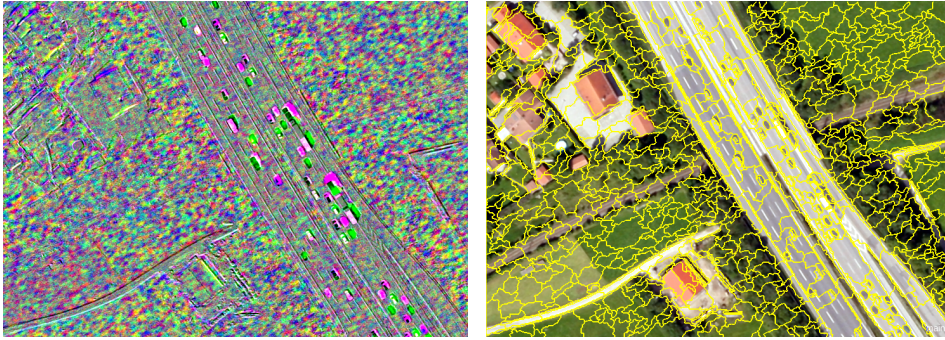
Die in der Verarbeitungsvariante C zur Segmentierung verwendeten Daten als auch die Segmentierung selbst werden in Abbildung 3.3 gezeigt.

### 3.2.4. Merkmalsextraktion

Nach der Segmentierung erfolgt die Merkmalsextraktion bei allen drei Verarbeitungsvarianten auf unterschiedliche Art. Die in Abschnitt 2.4 vorgestellten Methoden finden dabei Anwendung.

In Variante A wird die Methode der DCD genutzt, um die Segmentierung des ersten Aufnahmezeitpunkts  $S_1$  und die Segmentierung des zweiten Aufnahmezeitpunkts  $S_2$  zu verbinden. Da hierbei die Richtung der Objektentsprechung von Bedeutung ist, werden zwei Varianten erzeugt. Die Variante  $A_1$  ordnet jedem Objekt in  $S_1$  ein oder mehrere Objekte in  $S_2$  zu. Bei Variante  $A_2$  hingegen erfolgt diese Zuordnung in umgekehrter Richtung von  $S_2$  auf  $S_1$ . Daraus resultieren am Ende des Änderungsdetektionsprozesses zwei unterschiedliche Änderungsklassifikationen.





(a) MAD-Variante einer pixel-basierten IR-MAD-Transformation, die als Datengrundlage für die Segmentierung verwendet werden. (b) Gleichzeitige Darstellung beider Aufnahmen mit überlagertem Segmentierungsergebnis.

**Abb. 3.3.:** Eingabedaten und Ergebnis des MRS-Algorithmus auf Grundlage der pixel-basierten MAD-Variante für die Verarbeitungsvariante C.

Als Eigenschaften werden sowohl in Variante A<sub>1</sub> als auch in Variante A<sub>2</sub> die Mittelwerte, die Standardabweichungen und die GLCM Mittelwerte aller Kanäle verwendet. Zusätzlich wird die Objekteigenschaft *Compactness* benutzt. Damit werden insgesamt zehn Objektmerkmale für die Verarbeitungsvariante A verwendet.

Bei Variante B wird die Objektentsprechungsmethode CDIO eingesetzt. Damit werden die Segmentationen beider Aufnahmezeitpunkte  $S_1$  und  $S_2$  auf eine Segmentation  $S_{1,2}$  abgebildet. Die Informationen der Ausgangssegmentationen bleibt jedoch vollständig erhalten.

Zusätzlich zu den in der Variante A verwendeten Objekteigenschaften kommt in Variante B der sogenannte *Shape Index* zum Einsatz. Die Anzahl der verwendeten Merkmale beträgt somit elf.

Die Variante C schließlich benötigt keine Objektentsprechung, da nur eine Segmentation erzeugt wird.

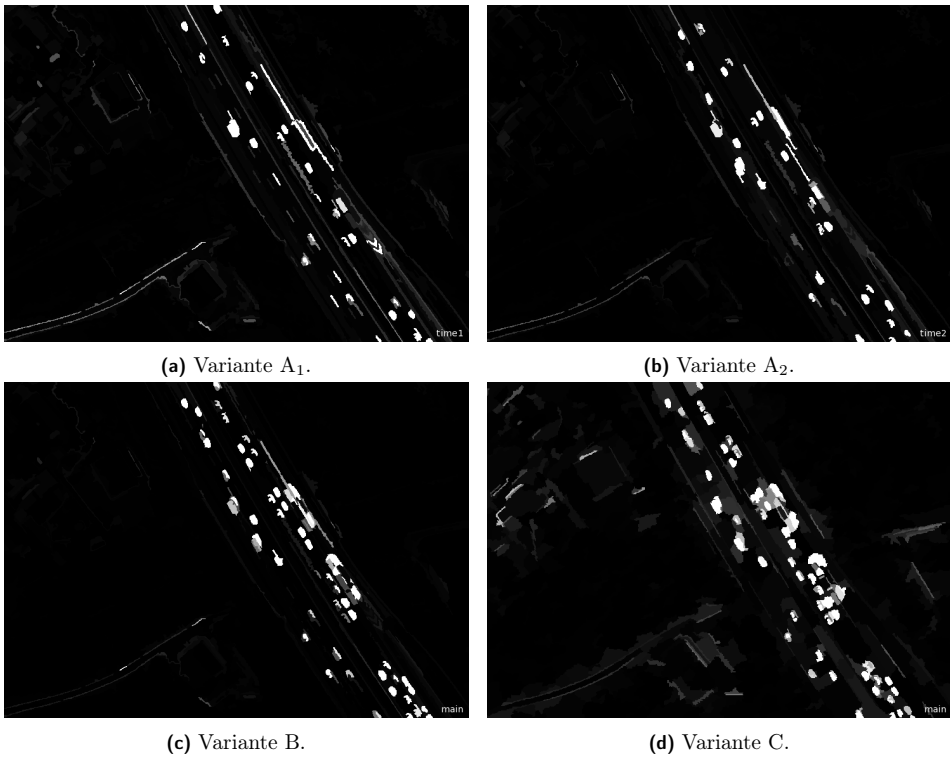
Als Objektmerkmale werden in der Variante C für alle Kanäle die Mittelwerte, die Standardabweichung, die mittleren Unterschiede im Mittelwert zu den Nachbarsegmenten, die GLCM Mittelwerte sowie die GLCM Homogenität verwendet. In Summe werden damit 15 Objekteigenschaften in Variante C verwendet.

### 3.2.5. Transformation des Merkmalsraumes

Um die Unterschiede zwischen den beiden Aufnahmen zu betonen, wird der Merkmalsraum der Objekte mit den Verfahren aus Abschnitt 2.5 transformiert.

Für alle drei Varianten findet zunächst eine Hauptkomponententransformation auf Grundlage der Daten beider Aufnahmezeitpunkte statt. Anschließend wird der IR-MAD-Algorithmus unter Eingabe der zuvor erzeugten Hauptkomponenten ausgeführt. Die Anzahl der verwendeten Hauptkomponenten bestimmt sich aus dem numerischen Verhalten des IR-MAD-Algorithmus, d.h. ob die Kovarianzmatrizen numerisch invertierbar sind. Es werden so viele Hauptkomponenten wie möglich, d.h. solange numerisch stabiles Verhalten vorliegt, benutzt. Für die Varianten A und B kommen daher drei Hauptkomponenten zur Anwendung. Bei Variante C können vier Hauptkomponenten in die Analyse einfließen.

Ein wesentliches Ergebnis der Merkmalsraumtransformation ist der Z-Wert des IR-MAD-Algorithmus. Dieses Ergebnis ist für alle Varianten der Verarbeitung in Abbildung 3.4 dargestellt.



**Abb. 3.4.:** Z-Werte der IR-MAD-Transformation für die einzelnen Varianten.



### 3.2.6. Änderungsklassifizierung

Auf Grundlage der MAD-Variate sowie dem Z-Wert aus der IR-MAD-Transformation kann eine Klassifizierung der Änderungen durchgeführt werden. Da in den Daten wesentliche Änderungen nur auf der Autobahn durch fahrende Fahrzeuge vorhanden sind, ergeben sich neben unveränderten Bereichen in der Klasse „no change“ nur zwei Arten von Änderungen: Bereiche, in denen sich während der ersten Aufnahme ein Auto befindet, in der zweiten Aufnahme jedoch nicht mehr (Klasse „time1:car, time2:road“) sowie Bereiche, in denen sich während der ersten Aufnahme kein Auto befindet, in der zweiten Aufnahme jedoch schon (Klasse „time1:road, time2:car“). Die dritte Art der Veränderung, der Fall dass sich in der ersten Aufnahme ein Auto und in der zweiten Aufnahme an derselben Stelle ein anderes Auto befindet, wurde wegen der geringen Anzahl an Beispielfällen innerhalb der Szene der Klasse „no change“ zugeordnet. Durch diese vorgegebene Definition der Änderungsarten kann für die Änderungsklassifizierung der Szene ein überwachtetes Klassifikationsverfahren angewendet werden. Zum Einsatz kommt ein Algorithmus unter Verwendung neuronaler Netzwerke, den *Class Based Neural Networks*, wie sie in Abschnitt 2.6.4 beschrieben wurden. Als Objektmerkmale finden die drei bzw. vier MAD-Variate sowie der Z-Wert Anwendung.

Im Detail wird für die beiden Varianten A<sub>1</sub> und A<sub>2</sub> der Algorithmus unabhängig voneinander ausgeführt, weil beide Ansätze unterschiedliche zeitliche Betrachtungsrichtungen für die Änderungen zugrundelegen. In A<sub>1</sub> wird daher auf Grundlage der Segmentation S<sub>1</sub> klassifiziert. Für A<sub>2</sub> wiederum erfolgt die Klassifizierung auf Basis der Segmentation S<sub>2</sub>. Die für A<sub>1</sub> verfügbaren Klassen sind somit durch „no change“ und „time1:car, time2:road“ gegeben. Die Klassen „no change“ und „time1:road, time2:car“ werden hingegen für die Variante A<sub>2</sub> benutzt. Dadurch erhält man zwei komplementäre Klassifikationen der Szene. Durch die Anzahl der verwendeten Klassen wird für den Parameter 1 **Hidden Layer** im Bereich **Class Network Setup** der Wert 2 eingesetzt. Für den Parameter 1 **Hidden Layer** im Bereich **Classification Network Setup** kommt der Wert 3 zum Einsatz. Als Parameterwert für den Lernalgorithmus in wird in beiden Varianten **Mixed (SCG+Kalman)** benutzt. Als Trainingsbeispiele werden in A<sub>1</sub> und A<sub>2</sub> für jede der verwendeten Klassen jeweils zehn Objekte durch den Nutzer ausgewählt.

In den Varianten B und C liegt nur eine Segmentation vor, auf der die Klassifikation ausgeführt wird. Dadurch werden alle drei Klassen „no change“, „time1:car, time2:road“ und „time1:road, time2:car“ gleichzeitig verwendet. Beide Varianten werden identisch im Prozessschritt der Änderungsklassifizierung identisch verarbeitet. Als Algorithmus kommt dasselbe Verfahren wie in Variante A zum Einsatz jedoch mit dem Parameterwert 3 für 1 **Hidden Layer** im Bereich **Class Network Setup** und den Wert 4 für den Parameter 1 **Hidden Layer** im Bereich **Classification Network Setup**. Ein weiterer Unterschied zu Variante A besteht in der Anzahl der Trainingsbeispielen. So werden durch den Nutzer für die Klassen „time1:car, time2:road“ und „time1:road, time2:car“ jeweils fünf und für die Klasse „no change“ sieben Objekte exemplarisch ausgewählt.

Die Klassifikationsergebnisse aller Verarbeitungsvarianten finden sich in Abbildung 3.5.

### 3.2.7. Genauigkeitsuntersuchung

Um die Güte der Änderungsklassifizierung zu bewerten, muss eine Genauigkeitsuntersuchung durchgeführt werden. Dazu wird das in Abschnitt 2.8.3 beschriebene Vorgehen angewendet. Als Referenzklassifizierung kommt zur Bewertung der Genauigkeit aller vier Varianten eine durch den Autor manuell erstellte Änderungsklassifizierung der Szene zum Einsatz (siehe Abbildung 3.6). Diese basiert auf den Pixeln des Bildes und ermöglicht somit eine integrierte Betrachtung der Qualität des Gesamtergebnisses. Die Ergebnisse der Güteuntersuchung finden sich in Tabelle 3.2.

Für die Varianten A<sub>1</sub> und A<sub>2</sub> werden durch die Art der Verarbeitung nur jeweils zwei Klassen pro Variante in der Klassifizierung eingesetzt. Daher findet die jeweils nicht vorhandene Klasse

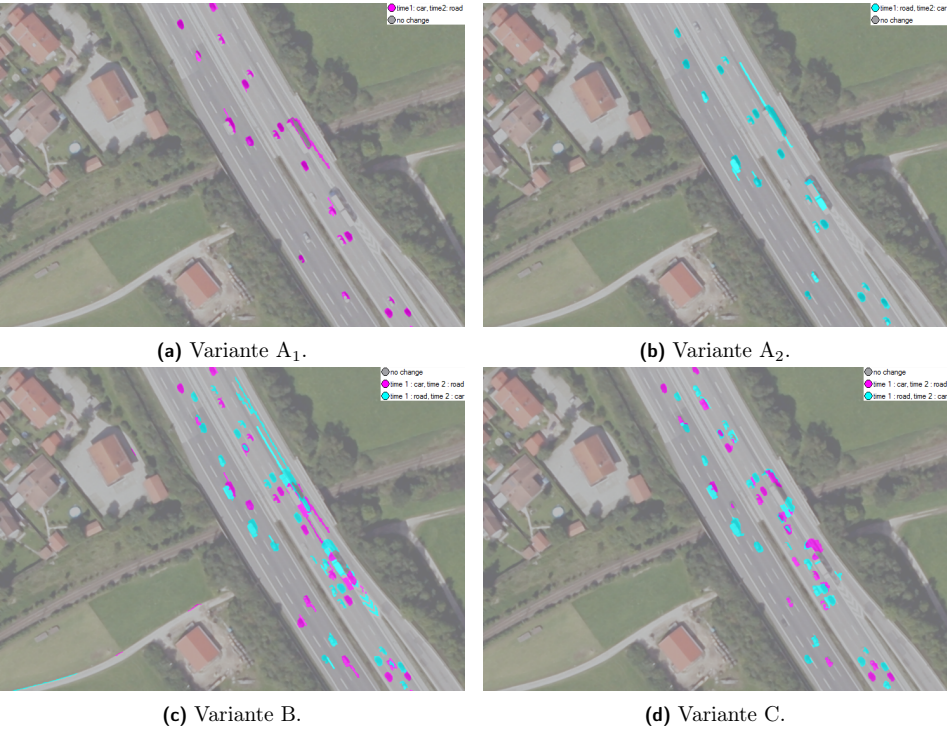


Abb. 3.5.: Klassifikationsergebnisse für die einzelnen Varianten.

Variante	A <sub>1</sub>		A <sub>2</sub>		B	C
Segmentierung	MRS4CD auf Grundlage der Bilddaten mit <i>Threshold Test</i> und <i>Universal Segment Removal</i> , Parameter: $w_s = 0$ in allen Phasen, $T = 50$ in Phase 1 und 5, $T_t = 58$ in Phase 3, $T = 25$ in Phase 6				MRS auf Grundlage der MAD-Variate, Parameter: $T = 3.5$ , $w_s = 0.01$ , $w_{comp} = 0.5$	
Objekt-entsprechung	DCD (T <sub>1</sub> ->T <sub>2</sub> )	DCD (T <sub>2</sub> ->T <sub>1</sub> )	CDIO		direkt	
Anzahl Merkmale	10				11	15
Transformation	PCA und IR-MAD auf Grundlage von 3 PCs				PCA und IR-MAD auf Grundlage von 4 PCs	
Änderungs-klassifizierung	CBNN auf Basis der MAD-Variate mit zehn Trainingsbeispielen pro Klasse				CBNN auf Basis der MAD-Variate mit jeweils fünf Samples für die Änderungsklassen und sieben für die Klasse der unveränderten Objekte	

Tab. 3.1.: Zusammenfassung der Verarbeitungsschritte.



Abb. 3.6.: Manuell erstellte Referenzklassifikation.

Variante	A <sub>1</sub>	A <sub>2</sub>	B	C
UA <sub>nochange</sub>	99.5%	99.6%	99.6%	99.6%
UA <sub>T1:car,T2:road</sub>	70.2%	N/A	64.6%	62.4%
UA <sub>T1:road,T2:car</sub>	N/A	69.2%	52.1%	67.6%
PA <sub>nochange</sub>	99.8%	99.7%	99%	99.7%
PA <sub>T1:car,T2:road</sub>	52.1%	N/A	67.1%	48.6%
PA <sub>T1:road,T2:car</sub>	N/A	64.2%	79.4%	70.8%
OA	99.3%	99.3%	98.5%	98.9%
$\widehat{K}$	59.4%	66.2%	67.3%	71.7%

Tab. 3.2.: Genauigkeit der Ergebnisse der Änderungsdetektion.

auch bei der Genauigkeitsuntersuchung keine Beachtung.

Es ist anzumerken, dass für die Genauigkeitsuntersuchung in Listner und Niemeyer (2011) die Genauigkeitsuntersuchung auf Grundlage der Objekte durchgeführt wurde. Daraus resultieren Unterschiede mit den im Rahmen dieser Arbeit vorgestellten Genauigkeitswerten in Tabelle 3.2.

### 3.2.8. Diskussion

Da für die Fallstudie „Autobahn“ alle Teilergebnisse im vorangegangenen Abschnitt präsentiert wurden, kann eine Diskussion dieser Ergebnisse im Detail für die drei Szenarien erfolgen. Hinsichtlich der Segmentierung wird deutlich, dass der MRS4CD-Algorithmus, der in Variante A und B zum Einsatz kommt, sich jeweils visuell sehr gut an die Daten der beiden Aufnahmezeitpunkte anpasst. Außerdem entstehen Änderungen in der Segmentation nur an Stellen, wo auch Änderungen in den Daten beobachtbar werden. Der Nutzung von Formeigenschaften im weiteren Prozessverlauf steht durch die Erzeugung zweier Segmentationen ebenfalls nichts entgegen. Allerdings ist die Parametrisierung des Algorithmus aufwändig. Dieser Aufwand ist bei Variante C durch die Verwendung einer einzigen Segmentation für beide Zeitpunkt nicht gegeben und ebenso die Verarbeitungszeit liegt bei Variante C deutlich unter jener der anderen Varianten. Die Anpassung der Segmentation an die Bilddaten ist in dieser Variante nicht so gut. Objekte, die sich geändert haben, extrahiert der Algorithmus hingegen mit hoher Qualität (siehe Abbildung 3.3b).

Die Objektentsprechung erfolgt in der Variante A mithilfe der DCD-Methode. Damit ist die Einbeziehung von Formeigenschaften in die Analyse möglich. Weil diese Methode jedoch gerichtete Verbindungen zwischen zwei Segmentationen erzeugt, entstehen die Variante A<sub>1</sub> für die eine zeitliche Betrachtungsrichtung und A<sub>2</sub> für die andere, was die Analyse zusätzlich

verkompliziert. Außerdem kann bei dieser Variante keine eindeutige Objektentsprechung garantiert werden und es muss somit im Falle von Mehrdeutigkeiten eine Mittelwertbildung erfolgen. Andererseits ist es auf diese Weise möglich, geänderte Bereiche auf Grundlage der ersten oder der zweiten Aufnahme darzustellen ohne eine neue „künstlich“ geschaffene Segmentation als Grundlage des Ergebnisses verwenden zu müssen. Dadurch ist es möglich die Bereiche mit verschwindenden bzw. neu auftauchenden Objekten auf der Fahrbahn zu markieren.

Variante B nutzt hingegen den CDIO-Ansatz, der ebenfalls die Integration von Formeigenschaften ermöglicht. Hierbei entsteht jedoch eine neue Segmentation, die mit den anderen Segmentationen verknüpft wird. Die Vorteile dieses Verfahrens liegen einerseits in der Erzeugung eines einzigen Endproduktes und der garantierten Eindeutigkeit der Objektverknüpfungen zu den Ausgangssegmentationen. Eine Neuberechnung der Merkmale wird somit nicht nötig. Jedoch können sich bei dieser Variante Änderungsobjekte überlagern wie man in Abbildung 3.5c sieht. Bei Variante C ist keine Objektentsprechung nötig, da nur eine Segmentation erzeugt wurde. Dadurch ist diese Variante hinsichtlich des Prozessschrittes der Objektentsprechung zu präferieren. Jedoch ist es aus diesem Grunde auch nicht möglich, Formeigenschaften in die Analyse einzubeziehen.

Hinsichtlich der Merkmalsraumtransformation gibt es keine Verarbeitungsunterschiede. Trotzdem differieren die Ergebnisse der einzelnen Varianten, da als Eingabe der Transformationen nicht dieselben Merkmale und Segmentationen verwendet wurden. Visuell überzeugen in Abbildung 3.4 vor allem die Varianten B und C. Dabei weist jedoch Variante C mehr Fehlindikatoren durch Fehler in der Koregistrierung auf als Variante B.

Hinsichtlich der Ergebnisse der überwachten Klassifikation in Abbildung 3.5 überzeugt Variante B hingegen nicht so sehr, da viele Bereiche als Änderung erkannt werden, die unverändert geblieben sind. Variante A hat den Vorteil, dass im Falle räumlich überlappender Änderungen, diese trotzdem korrekt dargestellt werden können. Als Gesamtprodukt der Änderungsklassifizierung überzeugt jedoch Variante C am meisten.

In der Genauigkeitsuntersuchung zeigen sich alle vier Varianten mit sehr hohen Gesamtgenauigkeiten in etwa gleichauf. Die hohe Genauigkeit ist jedoch der sehr kurzen Differenz zwischen den Aufnahmezeitpunkten geschuldet und sollte nicht überbewertet werden. In den Nutzer-genauigkeiten schneidet Variante A naturgemäß besser ab, da weniger Klassen betrachtet werden. Beim Kappaindex liegen jedoch Varianten B und C vorn.

Insgesamt gesehen ist keiner der vier Varianten ein eindeutiger Vorzug zu geben. Hinsichtlich der Genauigkeit sind die Varianten nahezu nicht unterscheidbar. Bei Zwischenergebnissen wie der Änderungsintensität ist hingegen die Variante B zu präferieren. Bei den Klassifikationsergebnissen sollte hingegen Variante A oder C gewählt werden.

Durch die Fallstudie „Autobahn“ konnte gezeigt werden, dass der Prozess nicht nur für simulierte, sondern auch für reale Daten handhabbar ist. Der komplizierte Segmentierungsprozess des MRS4CD-Algorithmus scheint jedoch keinen wesentlichen Vorteil gegenüber einer konventionellen Segmentierung zu bergen.

### 3.3. Fallstudie „Olkiluoto“

#### 3.3.1. Einleitung

Ein weiteres Anwendungsbeispiel soll anhand einer kerntechnischen Anlage in Finnland, dem Standort Olkiluoto, präsentiert werden. Als Bilddaten kommen zwei Quickbird-Aufnahmen vom 16. Juni 2005 und dem 17. Juli 2006 zum Einsatz (siehe Abbildung 3.7). Die Bilder wurden im Rahmen des LIMES-Projekts (siehe Gonçalves u. a. 2010) bezogen.

Die *Ground Sampling Distance* (GSD) beträgt 62 cm, woraus bei einem Bildformat von 1971 mal 893 Pixeln ein Gebiet von 1222 m mal 554 m auf der Erdoberfläche abgedeckt wird.

Zwischen den beiden Aufnahmezeitpunkten fanden an der Infrastruktur der Anlage zahlreiche Bauarbeiten statt, welche im visuellen Vergleich der Bilder gut sichtbar sind. Darüber hinaus treten jedoch in der Phänologie und dem Sonnenstand durch die saisonal und tageszeitlich ähnliche Aufnahmezeit kaum Unterschiede zutage. Für die Analyse anthropogener Änderungen sind die Daten daher sehr gut geeignet.

In den folgenden Abschnitten wird die Verarbeitung der Bilder im Detail beschrieben und die Ergebnisse diskutiert.



(a) Aufnahmezeitpunkt 16. Juni 2005.

(b) Aufnahmezeitpunkt 17. Juli 2006.

**Abb. 3.7.:** Die beiden zur Analyse verwendeten Satellitenaufnahmen.

### 3.3.2. Vorverarbeitung

In einem ersten Schritt werden beide Bilder koregistriert und panchromatisch geschärft. Da diese Verarbeitung bereits im Rahmen des LIMES-Projektes stattfand, liegen dem Autor über die konkreten Verfahrensschritte keine Informationen vor (siehe Gonçalves u. a. 2010, S. 43). Eine visuelle Qualitätskontrolle in der Bildverarbeitungssoftware *ENVI* zeigt jedoch, dass einerseits Effekte durch leicht unterschiedliche Aufnahmewinkel fast verschwinden. Andererseits besteht in nahezu allen Bildbereichen eine Genauigkeit der Ko-Registrierung, die Abweichungen von einem Pixel oder weniger aufweist. Auswirkungen der Pan-Schärfungen können visuell nicht detektiert werden.

Neben der Schaffung einheitlicher geometrischer Eigenschaften in beiden Bildern ist auch eine radiometrische Korrektur nötig. Dazu wird die Methode auf Grundlage der IR-MAD-Methode aus Canty u. a. (2004) verwendet. In der Regressionsphase des Verfahrens werden nur jene Pixel für die Schätzung der Parameter verwendet, deren Z-Wert nach der IR-MAD-Transformation auf eine Änderungswahrscheinlichkeit kleiner als 5% hinweist.

### 3.3.3. Segmentierung

Nach der Vorverarbeitung werden in einem zweiten Schritt die Bilder mit der Methode aus Abschnitt 2.3.4 segmentiert. Zunächst folgt die Bestimmung des Parameters  $T$ . Dazu wird das Bild des ersten Aufnahmezeitpunkts iterativ mit ansteigenden Werten für  $T$  segmentiert und für jeden Wert von  $T$  das Gütemaß  $E(S)$  berechnet. Es stellt sich heraus, dass im Wertebereich 20 bis 60 für  $T$  ein lokales Minimum für mehrere Werte von  $w_l$  bei  $T = 39$  existiert. Da das so erzielte Segmentierungsergebnis auch einer qualitativen Betrachtung standhält, wird dieser Wert für  $T$  im Folgenden für die Segmentierung in Phase 1 und 5 verwendet (siehe Tabelle 2.1). Für die Wichtung der Form werden die Parameter  $w_s = 0.2$  und  $w_{comp} = 0.8$  verwendet, da sich diese Parameterwahl bereits in John (2009) bewährt hat. Das Ergebnis der Parameterwahl für  $I_1$  ist in Abbildung 3.8 dargestellt.

Auf der Segmentation der ersten Aufnahme beruhend findet eine Anpassung an die zweite Aufnahme statt. Dabei stellt sich heraus, dass die Methoden des *Local Best Fitting Test* und *Local Mutual Best Fitting Test* zu wenig stabil für die vorliegenden Daten sind, d.h. subjektiv zu viele Segmente werden als mit der zweiten Aufnahme inkonsistent erklärt, bzw. die



**Abb. 3.8.:** Ausschnitt der Segmentation zum Aufnahmezeitpunkt 16. Juni 2005 mit  $T = 39$ ,  $w_s = 0.2$  und  $w_{comp} = 0.8$ .

Parameterwahl ( $T_t$ , **Mutual Tolerance** und **Minimum Tree Depth**) zu aufwändig wird. Daher werden die folgenden vier Kombinationen aus Testdaten und -methoden auf ihre Eignung für die Segmentierung der zweiten Aufnahme untersucht:

1. *Threshold Test* auf Grundlage der Bilddaten der zweiten Aufnahme,
2. *Relative Threshold Test* auf Grundlage der Bilddaten der zweiten Aufnahme,
3. *Threshold Test* auf Grundlage der MAD-Variate einer pixel-basierten IR-MAD-Transformation beider Aufnahmen,
4. *Threshold Test* auf Grundlage der Z-Werte einer pixel-basierten IR-MAD-Transformation beider Aufnahmen.

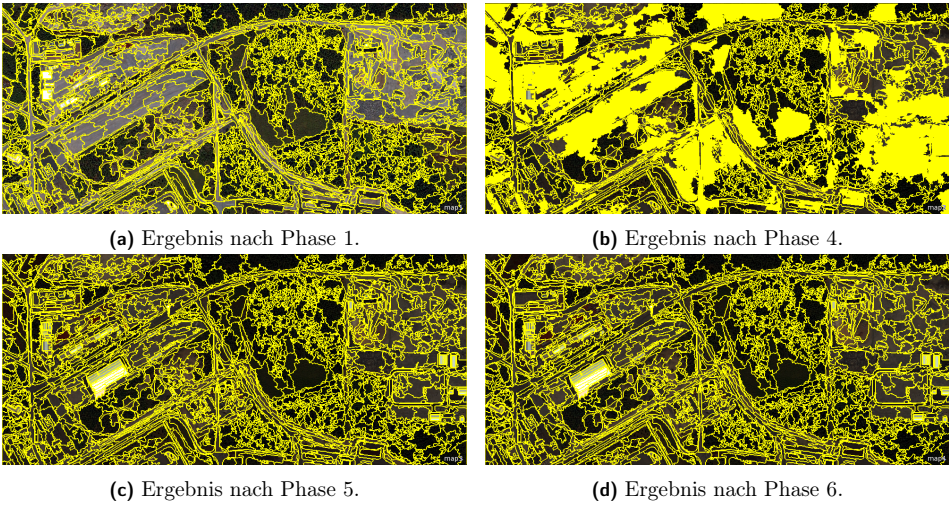
Konkret werden die Objekte der bereits in Abbildung 3.8 gezeigten Segmentation durch den Nutzer in zwei Klassen eingeteilt. Die erste Klasse NTBS, „not to be split“, soll jene Segmente umfassen, deren Form sich zwischen den beiden Aufnahmezeitpunkten nicht geändert hat und die daher auch nicht aufgeteilt werden sollen. Die zweite Klasse TBS, was für „to be split“ steht, soll hingegen jene Objekte beinhalten, deren Form sich wesentlich geändert hat. Da das MRS4CD-Plugin auch für jedes Segment eine Objektvariable `maxTValue` zuweist, kann anhand dieses Wertes mit der Methode aus 2.6.2 ein optimaler Schwellenwert  $T_t$  bestimmt werden. Zu beachten ist dabei, dass  $T_t$  so groß gewählt werden muss, dass keine Segmentaufteilung stattfindet. Um anschließend auch eine Genauigkeitsuntersuchung zu ermöglichen werden von den 125 Segmenten des Bildausschnitts 62 für die Schwellenwertbestimmung und 63 für die Genauigkeitsuntersuchung verwendet. Die Ergebnisse sind für die einzelnen Kombinationen aus Testdaten und -methoden in Tabelle 3.3 dargestellt.

Da der *Threshold Test* auf Grundlage des Z-Wertes mit Abstand die besten Ergebnisse zeigt, wird diese Testvariante gewählt. Darauf aufbauend wird die Methode zur Entfernung inkonsistenter Segmente gewählt. Dabei zeigt sich, dass die Methoden *Local Removal Strategy* und *Global Removal Strategy* für  $T_t = 251.56$  nicht adaptiv genug sind und ein kleinerer Wert für  $T_t$  wiederum zu viele Segmente aufteilt. Daher wird die Methode *Universal Segment Removal Strategy* eingesetzt. Die Ergebnisse des Segmentierungsschrittes sind in Abbildung 3.9 dargestellt.



	Variante 1	Variante 2	Variante 3	Variante 4
Datengrundlage	Bilddaten	Bilddaten	MAD-Variate	Z-Wert
Methode zur Konsistenzüberprüfung	<i>Threshold Test</i>	<i>Relative Threshold Test</i>	<i>Threshold Test</i>	<i>Threshold Test</i>
Schwellenwert	51.25	0	40.99	251.56
Verteilungsart (NTBS/TBS)	normal / lognormal	lognormal / lognormal	normal / normal	lognormal / lognormal
Gesamtgenauigkeit	83%	27%	73%	90%
Genauigkeit NTBS	100%	0%	100%	93%
Genauigkeit TBS	35%	100%	0%	82%
Bhattacharyya-Koeffizient	0.809	0.94	0.966	0.463

**Tab. 3.3.:** Ergebnisse der Schwellenwertauswahl für die bitemporale Segmentierung bei verschiedenen Bereinigungsmethoden und Datengrundlagen.



**Abb. 3.9.:** Segmentierung mithilfe des MRS4CD-Algorithmus.

### 3.3.4. Merkmalsextraktion

Im dritten Verarbeitungsschritt, der Merkmalsextraktion, werden zunächst die Segmentationen der beiden Aufnahmen mit der CDIO-Methode aus Abschnitt 2.4.2 verknüpft. Dazu werden zunächst beide Segmentationen als Grauwertbilder mit den Segment-IDs exportiert und in ein neues Projekt zusammen mit den Bilddaten importiert. Anschließend kann mit dem MRS-Algorithmus unter Verwendung der Parameter  $T = 0$  und  $w_s = 0$  auf Grundlage der Segment-ID-Ebenen die Segmentationen der einzelnen Zeitpunkte sowie eine Schnittmengensegmentation nach dem CDIO-Verfahren erstellt werden.

Zur Auswahl der für die weitere Analyse einzusetzenden Objekteigenschaften kommt die in Abschnitt 2.4.3 beschriebene Merkmalsentropie zur Anwendung. In Tabelle 3.4 sind die untersuchten Merkmale mit den jeweiligen Werten für die Entropie aufgeführt. Da zwischen den einzelnen Merkmalen kaum wesentliche Unterschiede bei der Entropie zu sehen sind, werden für die weitere Analyse alle Merkmale verwendet.

Merkmal	Entropie Aufnahme- zeitpunkt 1	Entropie Aufnahme- zeitpunkt 2	Durchschnitt der Entropie
GLCM Mean L4	10.1135	10.2084	10.16095
Mean L1	10.1132	10.2084	10.1608
GLCM Mean L2	10.1281	10.1889	10.1585
Mean L3	10.1153	10.1997	10.1575
GLCM Mean L1	10.1123	10.198	10.15515
GLCM Mean L3	10.1136	10.1935	10.15355
Mean L4	10.1033	10.1955	10.1494
Compactness	10.1164	10.1816	10.149
Mean L2	10.1087	10.1873	10.148
Std L2	10.1116	10.1835	10.14755
Std L3	10.1045	10.1838	10.14415
Std L1	10.1035	10.1791	10.1413
Shape index	10.0787	10.1926	10.13565
Std L4	10.0834	10.1876	10.1355
Area	9.55472	9.64205	9.598385

**Tab. 3.4.:** Objektmerkmale und zugehörige Entropiewerte absteigend sortiert nach dem Durchschnitt der Entropie.

### 3.3.5. Transformation des Merkmalsraumes

Der vierte Verarbeitungsschritt umfasst die Transformation des Merkmalsraumes mit den Methoden aus Abschnitt 2.5, so dass die Änderungen zwischen beiden Aufnahmen besser detektierbar werden. Dazu werden in den folgenden vier Unterabschnitten verschiedene Aspekte dieser Transformation untersucht.

Alle vier Untersuchungen betrachten in erster Linie die Änderungsintensität. Analysen zur Art der Änderungen finden sich in Abschnitt 3.3.6.

### Einfluss des Segmentierungsalgorithmus auf das Ergebnis der Änderungsdetektion

Um den Einfluss des Segmentierungsverfahrens auf die Ergebnisse der Merkmalsraumtransformation zu untersuchen, wird der IR-MAD-Algorithmus mit vorheriger Hauptkomponen-



tenanalyse auf die beiden Bilder angewandt. Anschließend soll das Ergebnis qualitativ begutachtet werden. Dazu kommen insgesamt fünf verschiedene Varianten der Bildsegmentierung zum Einsatz:

1. eine Segmentierung mit jeweils ein Pixel großen Objekten,
2. eine unabhängige Segmentierung beider Bilder und der Verwendung des CDIO-Verfahrens zur Objektentsprechung,
3. eine Segmentierung des ersten Bildes mithilfe des MRS-Algorithmus und die Nutzung derselben Segmentation für die zweite Aufnahme,
4. eine gemeinsame Segmentierung beider Aufnahmen mithilfe des MRS-Algorithmus,
5. eine Segmentierung beider Aufnahmen mithilfe des MRS4CD-Algorithmus.

Die verwendeten Merkmale umfassen die Mittelwerte und die Standardabweichungen aller Objekte in den Bildkanälen. Eine Ausnahme bildet die Segmentierung mit jeweils ein Pixel großen Objekten, bei der eine Berechnung der Standardabweichung keinen Sinn hat. Bei allen Segmentierungsvarianten wird dieser Merkmalsraum zunächst durch die Anwendung der Hauptkomponentenanalyse transformiert. Anschließend kommt der IR-MAD-Algorithmus zum Einsatz. Um Effekte durch die numerischen Probleme des IR-MAD-Verfahrens zu minimieren, werden höchstens so viele Hauptkomponenten als Eingabe verwendet, dass keine numerischen Probleme bei Anwendung des Plugins auftreten. Dies führt zur Verwendung von drei Hauptkomponenten in den Varianten 1-4. In Variante 5 kommen vier Hauptkomponenten zum Einsatz.

Die Z-Werte nach der IR-MAD-Transformation für die einzelnen Segmentierungsvarianten werden auf vier exemplarischen Bildausschnitten in Abbildung 3.11 dargestellt. Die dafür verwendeten Ausgangsdaten der Bildausschnitte werden in Abbildung 3.10 gezeigt.

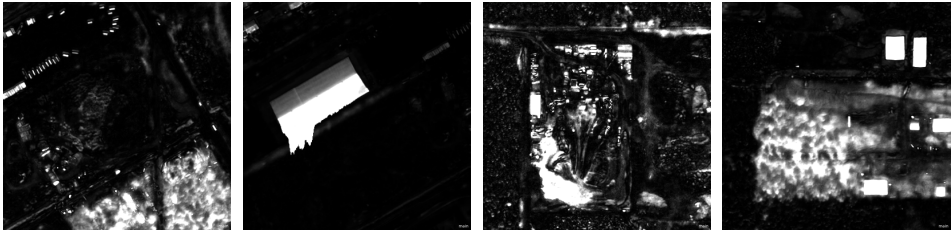


(a) Erster Aufnahmezeitpunkt.

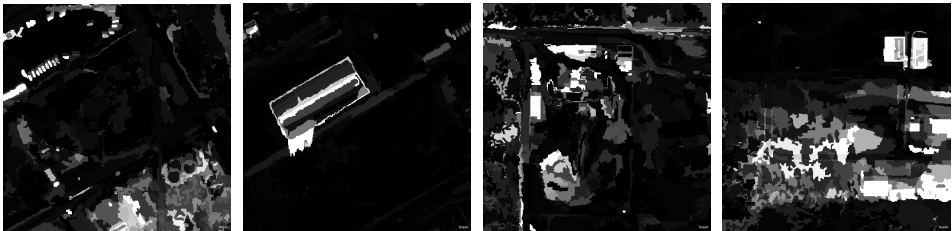


(b) Zweiter Aufnahmezeitpunkt.

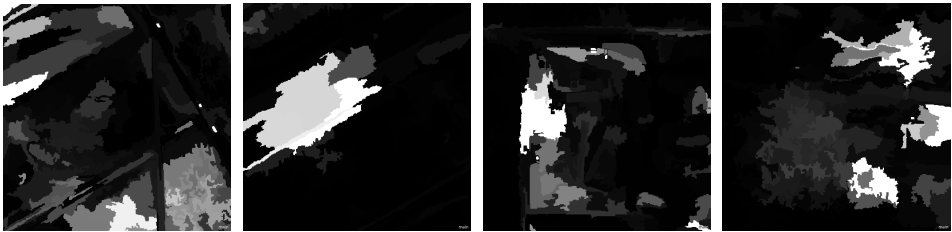
**Abb. 3.10.:** Vier exemplarische Ausschnitte des verwendeten Bilddatensatzes.



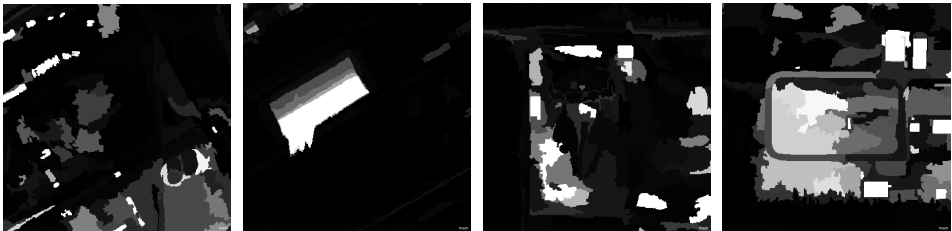
(a) Ergebnis des pixel-basierten Ansatzes gemäß Variante 1.



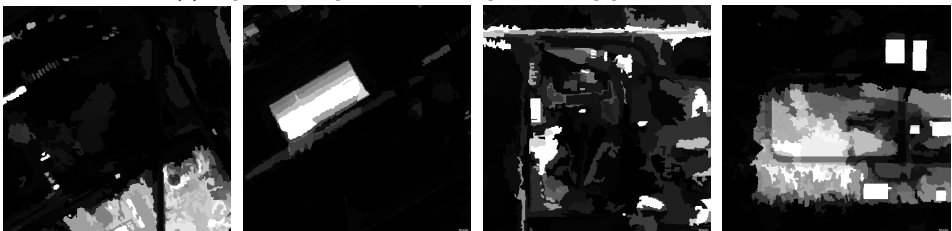
(b) Ergebnis der unabhängigen Segmentierung gemäß Variante 2.



(c) Ergebnis unter Nutzung der Segmentation des ersten Bildes gemäß Variante 3.



(d) Ergebnis der gemeinsamen Segmentierung gemäß Variante 4.



(e) Ergebnis des MRS4CD-Algorithmus gemäß Variante 5.

**Abb. 3.11.:** Ausschnitte der Änderungsintensitätsbilder bei Verwendung diverser Segmentierungsvarianten.

### Vergleich der iterativen Hauptkomponententransformation mit dem IR-MAD-Algorithmus

Neben einem Vergleich verschiedener Segmentierungsvarianten ist für den im Rahmen dieser Arbeit präsentierten Ansatz zur objekt-basierten Änderungsdetektion auch eine Betrachtung der Unterschiede zwischen verschiedenen Verfahren der Merkmalsraumtransformationen nötig. Dazu wird das Verfahren des IR-MAD-Algorithmus der iterativen Hauptkomponentenanalyse gegenübergestellt.

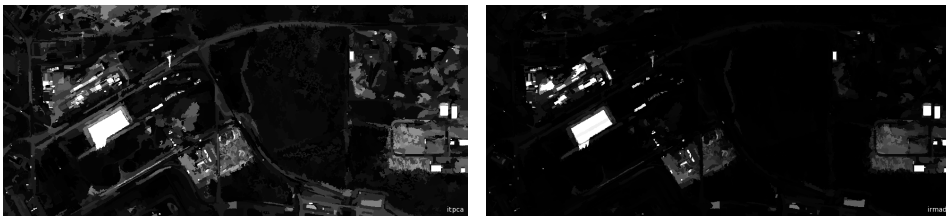
Die iterative Hauptkomponentenanalyse wird hierbei auf Grundlage der panchromatischen Intensität  $I$  ausgeführt. In der Realisierung wird die Intensität  $I$  eines Segments  $s$  durch die Objektmittelwerte im  $i$ -ten Kanal,  $\bar{x}_i(s)$ , berechnet. Dabei verwendet das Verfahren den Abstand zum Koordinatenursprung des Farbraumes, um die mittlere Intensität des Objekts zu bestimmen. Es gilt somit

$$I(s) = \sqrt{\sum_{i=1}^k (\bar{x}_i(s))^2}. \quad (3.1)$$

Für die IR-MAD-Transformation werden die Objektmittelwerte  $L_i(s)$  direkt verwendet, da ein multispektraler Datensatz für dieses Verfahren Voraussetzung ist.

Während für den Einsatz des IR-MAD-Verfahrens das in Abschnitt 2.5.4 beschriebene Plugin für *eCognition* direkt angewendet wird, muss für die iterative Hauptkomponentenanalyse ein mehrstufiger Ablauf unter Benutzung des PCA-Plugins aus Abschnitt 2.5.3 in *eCognition* implementiert werden. Im Detail wird zunächst der Merkmalsraum bestehend aus den Intensitätswerten der Segmente des ersten sowie des zweiten Aufnahmezeitpunkts transformiert. Dabei kommt jedem Segment dasselbe Gewicht zu. Anschließend wird mithilfe des FMLE-Algorithmus (siehe Abschnitt 2.6.3) anhand der zweiten Hauptkomponente eine Klassifikation in veränderte und unveränderte Objekte durchgeführt. Unveränderte Objekte werden dabei in einem Cluster  $NC$  in der Nähe des Nullpunktes vermutet. Veränderte Objekte treten entweder in einem Cluster  $C^+$  mit hohen positiven oder in einem Cluster  $C^-$  mit geringen negativen Werten auf. Daher wird der FMLE-Algorithmus mit dem Parameter  $c = 3$  ausgeführt. Der Cluster, dessen Betrag des Mittelwerts am geringsten ist, wird als Cluster der unveränderten Objekte betrachtet. Auf dieser Basis kann jedem Objekt der Wert der Zugehörigkeitsfunktion  $u$  für die Klasse  $NC$  als Wichtungsfaktor für die nächste Iteration zugewiesen werden.

Die Ergebnisse des Vergleichs der beiden Verfahren finden sich in Abbildung 3.12.



(a) Änderungsentensität ermittelt durch iterative Hauptkomponententransformation.

(b) Änderungsentensität ermittelt durch den IR-MAD-Algorithmus.

**Abb. 3.12.:** Indikatorbilder für Änderungen zwischen den beiden Aufnahmezeitpunkten.

### Auswirkungen des Einsatzes der Hauptkomponentenanalyse vor dem IR-MAD-Algorithmus

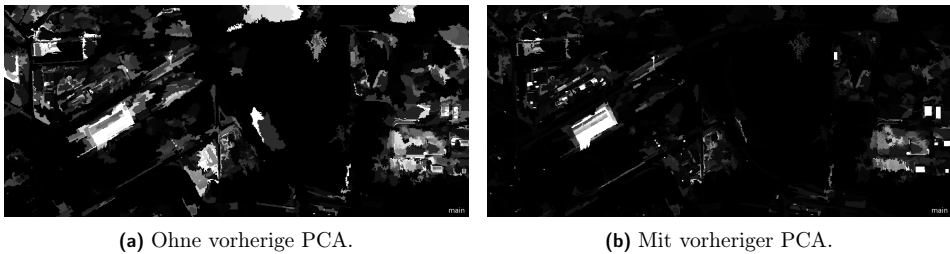
In Abschnitt 2.5.4 wurde aufgezeigt, dass der IR-MAD-Algorithmus aufgrund numerischer Probleme mit der Hauptkomponentenanalyse kombiniert werden sollte, um den Merkmals-

raum auf die wesentlichen Informationen zu reduzieren. Der Einfluss dieses Schrittes soll im Folgenden exemplarisch gezeigt werden.

Auf Grundlage des MRS4CD-Algorithmus, der CDIO-Methode zur Objektentsprechung und der in Tabelle 3.4 aufgeführten Objektmerkmale wird zunächst die Hauptkomponentenanalyse unter Verwendung der Daten beider Zeitpunkte ausgeführt. Anschließend werden die ersten drei Hauptkomponenten im IR-MAD-Algorithmus eingesetzt. Dieser Verarbeitung wird die direkte Anwendung des IR-MAD-Algorithmus auf die Merkmale aus Tabelle 3.4 gegenübergestellt.

Ohne vorherige Datenreduktion durch die Hauptkomponentenanalyse bricht der IR-MAD-Algorithmus nach acht Iterationen ab. Werden hingegen die Hauptkomponenten transformiert, so erfolgt eine numerisch stabile Verarbeitung in 21 Schritten.

Die Ergebnisse der beiden Herangehensweisen sind in Abbildung 3.13 dargestellt.



**Abb. 3.13.:** Z-Werte des IR-MAD-Algorithmus mit und ohne vorheriger Datenreduktion durch die Hauptkomponentenanalyse.

### Bedeutung der verschiedenen Objektmerkmale für die Änderungsanalyse

Abschließend soll im Rahmen der Betrachtung der Transformationsverfahren untersucht werden, welche Auswirkungen die Wahl der verwendeten Objektmerkmale auf das Ergebnis der Transformation hat. Als Segmentierungsverfahren kommt der MRS4CD-Algorithmus mit der CDIO-Methode zur Objektentsprechung zum Einsatz. Anschließend benutzt das Verfahren eine Hauptkomponentenanalyse, bevor der IR-MAD-Algorithmus verwendet wird.

Die in Tabelle 3.4 aufgeführten Objektmerkmale lassen sich in die Gruppen Mittelwert, Standardabweichung, Textur und Form einteilen. Die Untersuchung der Einflüsse der einzelnen Merkmalsgruppen auf das Ergebnis wird durchgeführt, indem sukzessive einzelne Gruppen von Merkmalen aus der Verarbeitung entfernt werden.

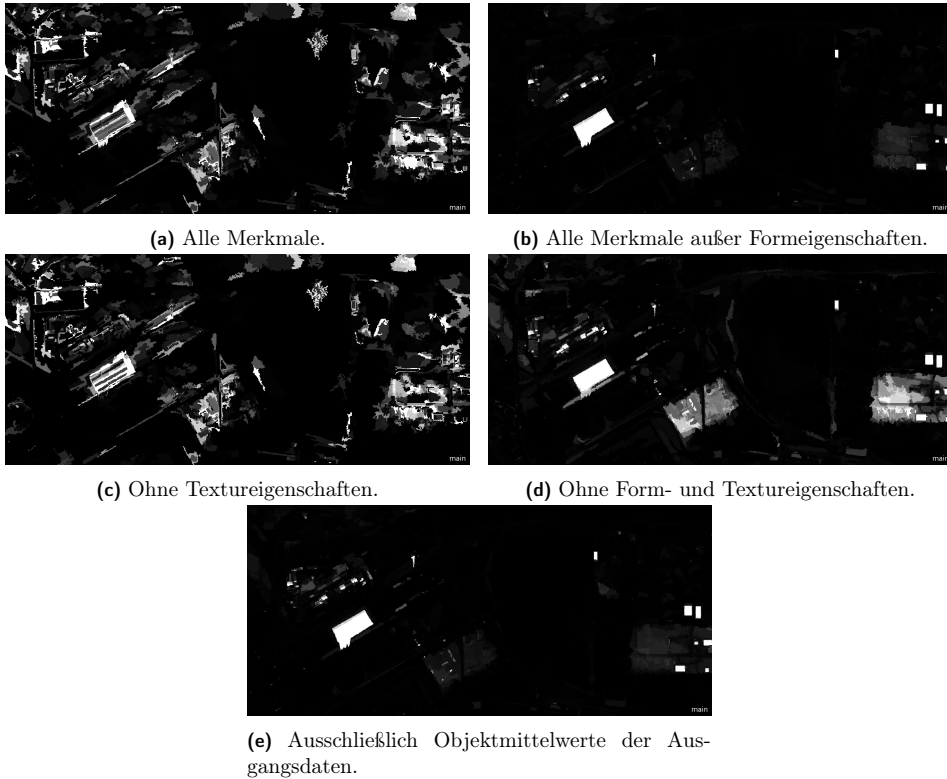
Die Ergebnisse sind in Abbildung 3.14 dargestellt.

#### 3.3.6. Änderungsklassifizierung

Die segmentierten und transformierten Bilddaten können in einem abschließenden Schritt mit den Methoden des Abschnitts 2.6 klassifiziert werden. Dazu kommen in den folgenden drei Teiluntersuchungen sowohl Schwellenwertverfahren als auch unüberwachte Klassifikatoren zum Einsatz.

#### Iterative FMLE-Klassifikation

Die erste Teiluntersuchung nutzt die mit dem MRS4CD-Algorithmus segmentierten, mit der CDIO-Methode kombinierten, der Hauptkomponentenanalyse und dem IR-MAD-Algorithmus transformierten Daten unter Einsatz der Objektmittelwerte und -standardabweichungen zur



**Abb. 3.14.:** Z-Werte des IR-MAD-Algorithmus unter Verwendung verschiedener Objektmerkmale.

Änderungsklassifikation. Es werden dabei sowohl Änderungsgruppen gebildet, als auch unveränderte Objekte ausgefiltert.

Im Detail wird in mehreren Durchläufen der FMLE-Algorithmus mit  $c = 6$  auf Grundlage der MAD-Variate ausgeführt. Anschließend werden die Objekte jenes Clusters von der weiteren Analyse ausgeschlossen, deren mittlerer  $Z$ -Wert unter allen Clustern minimal ist. Die Iteration bricht ab, sobald gemäß des Verfahrens eine Klasse mit relevanten Änderungen von der weiteren Analyse ausgeschlossen werden müsste. Diese Einschätzung ist durch den Analysten zu treffen und bedarf einer qualitativen Einschätzung.

Die Ergebnisse des Verfahrens sind bis zur sechsten Iteration in Abbildung 3.15 dargestellt. Während der sechsten Iteration werden aus Sicht des Autors relevante Änderungen entfernt, weshalb das Verfahren an dieser Stelle abbricht. Als Ergebnis des Verfahrens wird somit die Klassifikation der sechsten Iteration vor der Löschung angesehen (siehe Abbildung 3.15k).

### Schwellenwertklassifikation

Die zweite Teiluntersuchung zur Änderungsklassifizierung führt eine Schwellenwertklassifikation anhand der in Abschnitt 3.3.5 erstellten Transformationen für die einzelnen Segmentierungsvarianten durch. Um den Schwellenwert mithilfe der Methode aus Abschnitt 2.6.2 zu bestimmen, werden 100 zufällig auf der Fläche des Bildes gleichverteilte Pixel als Proben genommen und durch den Autor in die Klassen „Verändert“ (33 Stück) und „Unverändert“ (67 Stück) eingeteilt. Von diesen 100 Proben werden 50 zur Bestimmung des Schwellenwertes und 50 zur Untersuchung der Genauigkeit verwendet (siehe Abschnitt 3.3.7).

Zur Bestimmung des Schwellenwertes wird für jedes Pixel der  $Z$ -Wert des zugehörigen Objekts der Segmentationen über die Segmentationshierarchie ermittelt und in der Analyse verwendet. Für die Genauigkeitsuntersuchung wird die Klassenzugehörigkeit eines Pixels ebenfalls über die Hierarchie der Segmentationen bestimmt.

Die Klassifikationsergebnisse sind in Abbildung 3.16 dargestellt. Die Untersuchung der Genauigkeit findet in Abschnitt 3.3.7 statt.

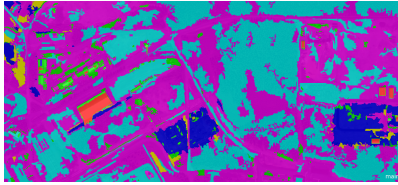
### FMLE-Klassifikation mit Schwellenwertfilterung

Die dritte Teiluntersuchung verbindet die Methode der Schwellenwertklassifikation aus Abschnitt 2.6.2 mit dem Ansatz der unüberwachten Klassifizierung durch den FMLE-Algorithmus aus Abschnitt 2.6.3. Dazu wird zunächst ein Schwellenwert auf den  $Z$ -Wert der IR-MAD-Transformation angewandt. Anschließend werden die Segmente, deren  $Z$ -Wert oberhalb des Schwellenwertes liegt, mit dem FMLE-Algorithmus in Gruppen gleichartiger Änderungen eingeteilt.

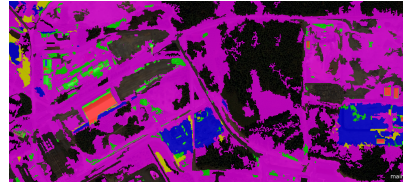
Als Grundlage wird das Transformationsergebnis aus Abschnitt 3.3.5 gemäß MRS4CD-Segmentierungsvariante 5 mit dem Schwellenwert 6346694 auf den  $Z$ -Wert benutzt.

Ziel der Untersuchung ist es, die optimale Clusteranzahl  $c$  zu bestimmen. Dazu wird der FMLE-Algorithmus mehrfach für alle Werte von  $c$  zwischen 2 und 50 ausgeführt und jeweils die Gütemaße *Fuzzy Hypervolume*,  $F_{HV}$ , und *Partition Density*,  $P_D$ , aus Abschnitt 2.6.3 berechnet. Die Werteverläufe sind in Abbildung 3.18 dargestellt. Die Optima gemäß der erläuterten Erweiterung des Optimalitätsbegriffs sind in rot dargestellt.

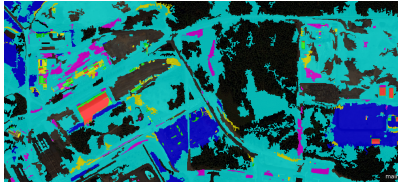
Es fällt auf, dass viele lokale Optima, d.h. Minima in Abbildung 3.17a und Maxima in Abbildung 3.17b, in den Daten erscheinen. Man kann den Gedanken des lokalen Optimums jedoch ausweiten, so dass ein Punkt optimal ist, wenn die vier davor und vier danach liegenden Datenpunkte jeweils größer bzw. kleiner als ein optimaler Punkt sind (rote markierte Datenpunkte in Abbildung 3.17b). Dann bleiben für  $c < 20$  noch zwei Optima bei  $c = 6$  und  $c = 15$ . Das Ergebnis für diese beiden Werte des Parameters  $c$  ist in Abbildung 3.18 dargestellt. Die Begrenzung von  $c < 20$  wird dabei gewählt, um die Anzahl der Klassen in einem für den Analysten noch überschaubaren Rahmen zu halten.



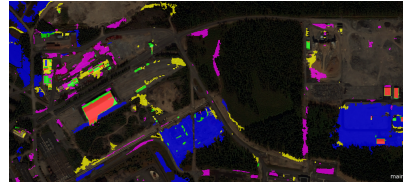
(a) Erste Iteration.



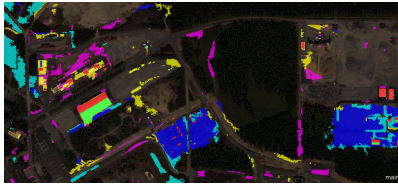
(b) Erste Iteration nach Löschung.



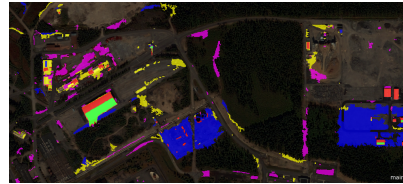
(c) Zweite Iteration.



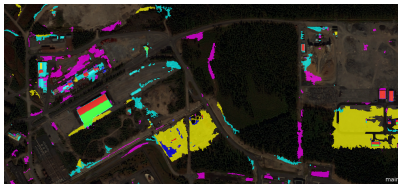
(d) Zweite Iteration nach Löschung.



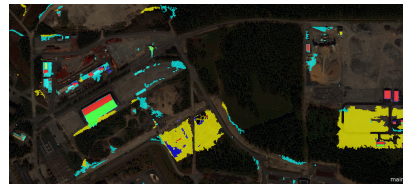
(e) Dritte Iteration.



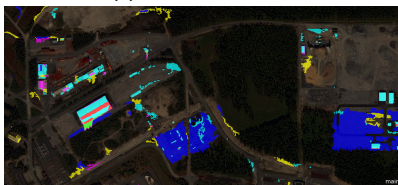
(f) Dritte Iteration nach Löschung.



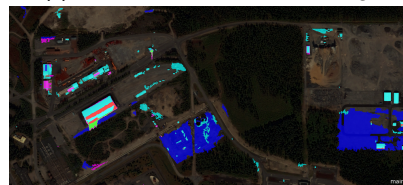
(g) Vierte Iteration.



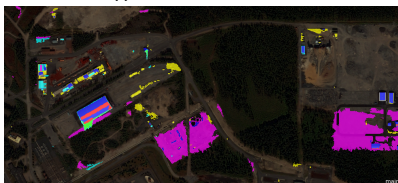
(h) Vierte Iteration nach Löschung.



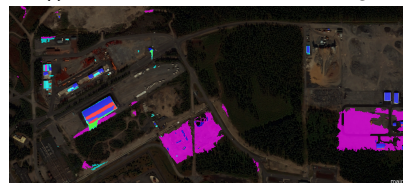
(i) Fünfte Iteration.



(j) Fünfte Iteration nach Löschung.



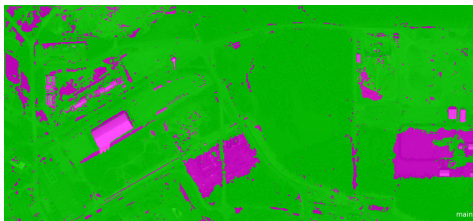
(k) Sechste Iteration.



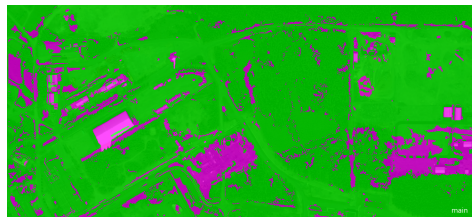
(l) Sechste Iteration nach Löschung.

**Abb. 3.15.:** Ergebnis der iterativen Anwendung des FMLE-Algorithmus bei Entfernung des Clusters mit den durchschnittlich geringsten Z-Werten.

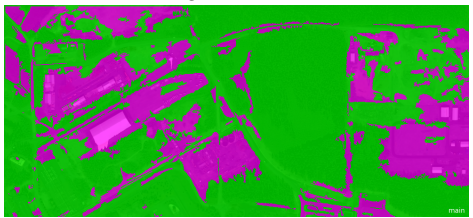




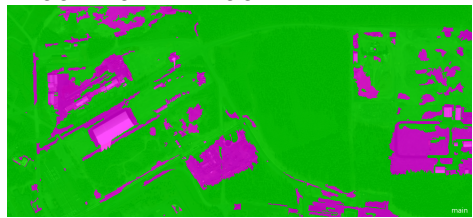
(a) Klassifikationsergebnis bei Nutzung des pixelbasierten Ansatzes gemäß Variante 1.



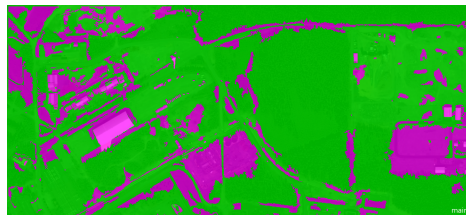
(b) Klassifikationsergebnis bei Nutzung der unabhängigen Segmentierung gemäß Variante 2.



(c) Klassifikationsergebnis bei Nutzung der Segmentierung der ersten Aufnahme gemäß Variante 3.



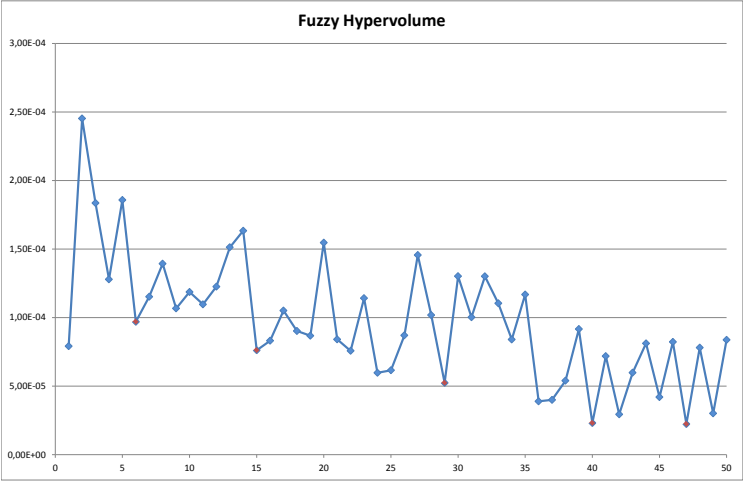
(d) Klassifikationsergebnis bei Nutzung der gemeinsamen Segmentierung gemäß Variante 4.



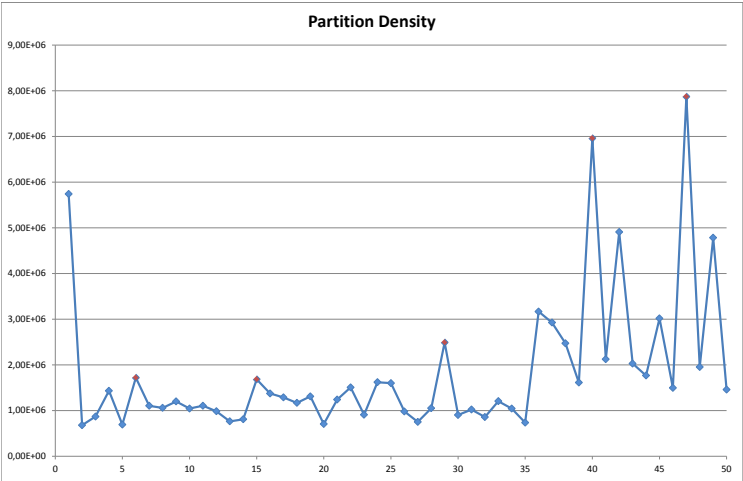
(e) Klassifikationsergebnis bei Nutzung des MRS4CD-Algorithmus gemäß Variante 5.

**Abb. 3.16.:** Ergebnisse der Änderungsklassifikation mit Schwellenwertverfahren.



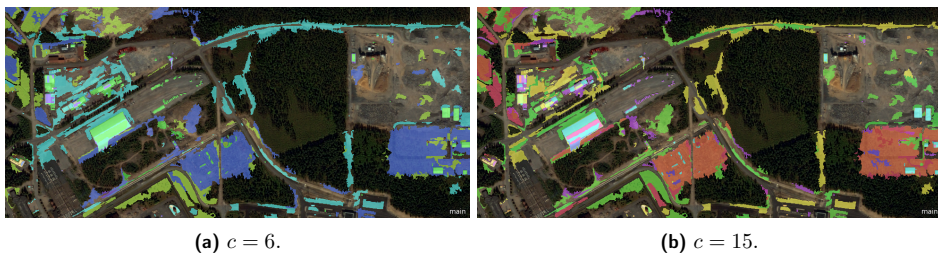


(a) *Fuzzy Hypervolume*  $F_{HV}$ .



(b) *Partition Density*  $P_D$ .

**Abb. 3.17.:** Verlauf der Gütemaße in Abhängigkeit von der Clusteranzahl  $c$ .



**Abb. 3.18.:** Ergebnis des FMLE-Algorithmus für verschiedene Werte des Parameters  $c$ .

### 3.3.7. Genauigkeitsuntersuchung

Um die Ergebnisse der Änderungsklassifizierung zu bewerten, wird eine Untersuchung der Genauigkeit gemäß der in Abschnitt 2.8.3 beschriebenen Methode durchgeführt. Dazu kommen 100 jeweils ein Pixel große Proben zum Einsatz, die durch den Autor manuell den Klassen „Verändert“ und „Unverändert“ zugeordnet werden. Ob eine Pixelprobe durch den jeweiligen Algorithmus korrekt klassifiziert wird, kann über die Klassenzugehörigkeit des Objekts bestimmt werden, in dessen Bereich sich das Pixel befindet.

Für die Gütebetrachtung der iterativen FMLE-Klassifikation aus Abschnitt 3.3.6 werden alle 100 Proben verwendet. Da der Algorithmus nach der fünften Iteration abbricht, dient das so erzielte Ergebnis als Ausgangspunkt für die Genauigkeitsuntersuchung.

In Tabelle 3.5 sind die Kennzahlen für die Genauigkeitsuntersuchung dargestellt. Die Gesamtgenauigkeit erreicht dabei nach der fünften Iteration einen Wert von 75%.

Die Betrachtung der Ergebnisqualität des Schwellenwertansatzes aus Abschnitt 3.3.6 erfolgt

Iteration	1	2	3	4	5	6
<b>PA<sub>Verändert</sub></b>	93.94%	39.39%	27.27%	24.24%	24.24%	18.18%
<b>PA<sub>Unverändert</sub></b>	62.69%	98.51%	100%	100%	100%	100%
<b>UA<sub>Verändert</sub></b>	55.36%	92.86%	100%	100%	100%	100%
<b>UA<sub>Unverändert</sub></b>	95.45%	76.74%	73.63%	72.83%	72.83%	71.28%
<b>OA</b>	73%	79%	76%	75%	75%	73%
<b><math>\hat{K}</math></b>	48.12%	44.39%	33.44%	30.01%	30.01%	22.95%

**Tab. 3.5.:** Genauigkeit der Ergebnisse nach Verarbeitung mit dem iterativen FMLE-Algorithmus.

ebenfalls unter Verwendung der 100 Proben. Allerdings wird die Hälfte der Proben zur Schwellenwertbestimmung und die andere Hälfte zur Güteuntersuchung benutzt. Die Auswahl der Proben erfolgt dabei zufällig, wobei für alle Varianten dieselbe Auswahl Verwendung findet. In Tabelle 3.6 sind die Ergebnisse der Analyse für die einzelnen Segmentierungsvarianten dargestellt.

Segmentierungsvariante	1	2	3	4	5
<b>PA<sub>Verändert</sub></b>	47.06%	35.29%	68.75%	41.18%	43.75%
<b>PA<sub>Unverändert</sub></b>	100%	93.94%	94.12%	100%	91.18%
<b>UA<sub>Verändert</sub></b>	78.57%	73.81%	86.49%	76.74%	77.5%
<b>UA<sub>Unverändert</sub></b>	100%	75%	84.62%	100%	70%
<b>OA</b>	82%	74%	86%	80%	76%

**Tab. 3.6.:** Genauigkeit der Ergebnisse nach Verarbeitung mit der Schwellenwertklassifikation.

### 3.3.8. Diskussion

Analog zum Unterabschnitt 3.2.8 kann auch für die Anwendung „Olkiluoto“ die Diskussion im Detail für die fünf verwendeten Szenarien erfolgen. Da die Daten jedoch durch unterschiedliche Störeinflüsse beeinträchtigt sind, kann die Qualität der Ergebnisse aus diesem Abschnitt nur bedingt mit jenen der Fallstudie aus Abschnitt 3.2 verglichen werden.

In der Segmentierungsstufe des Prozesses kann beobachtet werden, dass die Methoden *Global Removal Strategy* und *Local Removal Strategy* bei dem als optimal identifizierten Schwellenwert von  $T_t = 251.56$  nicht ausreichend an die Daten des zweiten Aufnahmezeitpunkts anpassen. Eine Reduzierung des Schwellenwertes teilt wiederum zu viele Segmente auf. Ein möglicher Ausweg wäre die Markierung aufzuteilender Segmente nicht auf Grundlage der kompletten Segmente des ersten Aufnahmezeitpunkts, sondern nur für Teile. Dazu müsste jedoch die Implementierung des MRS4CD-Algorithmus wesentlich angepasst werden. Für die Segmentierung mit der *Universal Removal Strategy* gilt außerdem, dass diese Methode zwar viele Segmente aufteilt, diese jedoch auch wieder zusammenfügt und so im Endeffekt trotzdem nur geringe Änderungen zwischen den Segmentationen entstehen. Zudem liegt ein wichtiges Resultat darin, dass als Datengrundlage der Konsistenzüberprüfung auch die Intensität einer pixel-basierten IR-MAD-Transformation verwendet werden können.

Bei der Merkmalsextraktion zeigt sich in dieser Fallstudie, dass die CDIO-Methode gegenüber dem DCD-Ansatz im Vorteil ist, weil die Bereiche der Änderung so besser in Ihrer Form dargestellt werden und somit die Interpretation erleichtert wird. Der in Abschnitt 3.2 gesehene Vorteil des DCD-Ansatzes trifft hier aufgrund der vielfältigen Arten von Änderungen nicht zu.

Bei der Transformation des Merkmalsraumes können im Rahmen der vorliegenden Fallstudie vier Teiluntersuchungen diskutiert werden. Die erste Teiluntersuchung, deren Ergebnis als Änderungsintensität in Abbildung 3.11 dargestellt ist, zeigt die diversen Vor- und Nachteile der unterschiedlichen Segmentierungsansätze. Während bei einem pixel-basierten Ansatz in Variante 1 eine große Detailfülle vorhanden ist, führt dieses Verfahren auch zu einer gewissen Unübersichtlichkeit, da es sich nicht auf wesentliche Änderungen beschränkt. Variante 2 zeigt hingegen die Schwächen der unabhängigen Segmentierung der beiden Aufnahmen, da hierbei in der Transformation viele Objekte mit großen Änderungsintensitäten auftreten, die aus geringer Überlappung zwischen instabil extrahierten Objekten resultieren. Variante 3 zeigt im Gegensatz dazu, wie eine Segmentation, die nicht an die zweite Aufnahme angepasst ist, das Änderungsergebnis verfälschen kann. Die großen Objekte, welche für die erste Aufnahme plausibel sind, ermöglichen so keine sinnvolle Interpretation der Änderungen. Variante 4 zeigt zwar insgesamt gute Ergebnisse, jedoch werden flächenhaften Änderungen geringer Intensität nicht ausreichend betont. Variante 5 zeigt schließlich die am leichtesten zu interpretierende Karte der Änderungsintensität. Weder erscheinen zu viele Details und Fehlalarme, noch werden wesentliche Änderungen verpasst.

Die zweite Teiluntersuchung im Rahmen der Merkmalsraumtransformation vergleicht die Änderungsintensität auf Grundlage der iterativen PCA mit jener des IR-MAD-Algorithmus (siehe Abbildung 3.12). Dabei ist festzustellen, dass die PCA flächenhafte Änderungen besser identifiziert. Jedoch gehen dadurch im Vergleich zum IR-MAD-Algorithmus kleine meist anthropogene Änderungen leicht unter. Die iterative PCA hat zudem den Nachteil, dass keine multivariate Verarbeitung möglich ist und somit die Vorteile des objekt-basierten Ansatzes nur bedingt genutzt werden können.

In der dritten Teiluntersuchung wird der Einfluss der Datenverarbeitung und Merkmalsreduktion mittels PCA auf das Ergebnis des IR-MAD-Algorithmus analysiert. Durch die PCA wird zudem eine numerische Stabilisierung des IR-MAD-Algorithmus erreicht. Die Ergebnisse in Abbildung 3.13 zeigen deutlich, dass eine numerische Stabilisierung der IR-MAD-Methode zu besseren Ergebnissen führt.

Schließlich wird in der vierten Teiluntersuchung der Einfluss der unterschiedlichen Eigenschaftsarten auf das Ergebnis untersucht. Qualitativ kann aus Abbildung 3.14 abgeleitet werden, dass Form- und Textureigenschaften in der hier diskutierten Fallstudie nicht zu einer Verbesserung der Ergebnisse beitragen. Die ausschließliche Verwendung der Kanalmittelwerte ist jedoch auch nicht empfehlenswert. Die besten Ergebnisse treten beim Einsatz von Kanalmittelwerten und Standardabweichungen zutage.

Hinsichtlich der Klassifikation mithilfe des iterativen FMLE-Algorithmus werden im Rahmen dieser Fallstudie qualitativ sehr gute Ergebnisse erzielt (siehe Abbildung 3.15), die nicht nur eine Unterscheidung nach Änderungsintensität, sondern auch nach -typ erlauben. Der Preis dafür ist jedoch der hohe Grad an Nutzerinteraktion.

Alternativ kann mithilfe der automatisierten Schwellenwertbestimmung ein Verfahren zum Einsatz kommen, das auf Basis vorher gewählter Trainingsgebiete ausgeführt wird. Dieses Verfahren liefert in Abhängigkeit vom Segmentierungsalgorithmus ausreichende Ergebnisse. Zwar gehören hier die Ergebnisse des MRS4CD-Algorithmus zu den besseren. Allerdings kann der Aufwand im Vergleich zur Segmentierung beider Bilder als ein Datensatz nicht durch die Ergebnisse gerechtfertigt werden.

Bei der Kombination des FMLE-Algorithmus mit dem Verfahren der schwellenwert-basierten Klassifikation hat die Fallstudie gezeigt, dass die optimale Wahl der Klassen selbst unter Einbeziehung der Gütemaße aus Abschnitt 2.6.3 nicht trivial ist.

Die Genauigkeitsuntersuchung der Klassifikationen zeigt schließlich, dass die Wahl des Segmentierungsalgorithmus die Genauigkeit bestimmt. Der Mehraufwand der Nutzung des MRS4CD-Algorithmus kann jedoch nur auf Grundlage der Genauigkeit nicht gerechtfertigt werden. Für alle Segmentierungsvarianten sind die Genauigkeiten zudem wesentlich geringer als in der Fallstudie aus Abschnitt 3.2. Die höhere Qualität bei der visuellen Betrachtung der Klassifikationsergebnisse und insbesondere der Änderungsintensitäten widerspricht dieser Schlussfolgerung allerdings. Daher sollte die Genauigkeit der Klassifikation nicht als alleinige Maßzahl zur Bewertung der Qualität einer Änderungsanalyse angesehen werden.

### 3.4. Zusammenfassung der Ergebnisse

Die Ausführungen dieses Kapitels haben gezeigt, dass die vollständig objekt-basierte Methode zur Änderungsdetektion auch in praktischen Anwendungen einsetzbar ist. Dabei sind jedoch einige Punkte bei der Implementierung des Prozesses zu beachten, die im Folgenden zusammenfassend aufgeführt werden sollen.

Die Vorverarbeitung spielt eine große Rolle für die Qualität des Endergebnisses. Insbesondere die geometrische Qualität der Daten ist hierfür wichtig. Bezüglich der radiometrischen Eigenschaften genügt die Verwendung eines relativen Korrekturverfahrens.

Der Segmentierungsschritt bildet die Grundlage der objekt-basierten Änderungsdetektion. Mithilfe des MRS4CD-Algorithmus kann eine Segmentation erzielt werden, die sich einerseits gut an die Daten beider Aufnahmezeitpunkte anpasst, aber kaum Unterschiede in Bildbereichen ohne relevante Änderung aufweist. Dazu sollte in der Praxis vorzugsweise der *Threshold Test* in Kombination mit der *Universal Removal Strategy* zum Einsatz kommen. Der Parameter  $T_t$  des MRS4CD-Algorithmus kann dabei elegant mithilfe einer schwellenwert-basierten Klassifikation ermittelt werden.

In der Phase der Objektentsprechung ist die Frage nach dem zu verwendenden Verfahren nur in Abhängigkeit vom jeweiligen Fall zu beantworten. Sind hauptsächlich Änderungen zwischen den beiden Aufnahmen zu beobachten, die auf die Verschiebung von kleineren Objekten in Bildbereiche mit ursprünglich größeren Objekten zurückzuführen sind, so kann die DCD-Methode eingesetzt werden. In allen anderen Fällen sollte dem CDIO-Verfahren der Vorzug gegeben werden.

Hinsichtlich der Merkmalsraumtransformation kann der IR-MAD-Algorithmus gegenüber der iterativen PCA mit einer Reihe von Vorzügen punkten. Numerische Instabilitäten bei der Verwendung vieler untereinander stark korrelierter Objekteigenschaften sind jedoch beobachtbar. Daher empfiehlt sich die vorherige Reduktion der Merkmalsanzahl z.B. mithilfe der PCA. Eine Karte der Änderungsintensitäten kann, obwohl sie eigentlich nur ein Zwischenergebnis darstellt, für den Analysten sehr hilfreich sein.

Bei der Klassifikation sind sowohl der schwellenwert-basierte Ansatz, die unüberwachte FMLE-Methode als auch die überwachte Klassifikation mit neuronalen Netzwerken in der Praxis einsetzbar. Je nach Anwendungsfall ist dem einen Verfahren vor dem anderen der Vorzug zu geben. Unter anderem hängt die Wahl davon ab, ob ein Modell der potentiellen Änderungsarten existiert. Auch hat die Frage nach dem gewünschten thematischen Detaillierungsgrad der Änderungskarte einen Einfluss auf diese Entscheidung. Eine generelle Empfehlung, die auf alle Anwendungsfälle zutrifft, ist daher nicht möglich.

Zur Genauigkeitsuntersuchung ist zu sagen, dass eine einzelne Zahl für die Gesamtgenauigkeit die Qualität des Ergebnisses einer Änderungsdetektion nur bedingt wiedergibt. Eine teilweise qualitative Analyse der Ergebnisse ist daher ratsam.

Allgemein besitzt die Methode viele Stellschrauben, die für die jeweiligen Gegebenheiten aufwändig angepasst werden müssen. Die vorliegende Arbeit versucht jedoch dem Nutzer Werkzeuge an die Hand zu geben, um diese Herausforderung zu bewältigen. Beispiele dafür sind die Methoden zur Parameterbestimmung beim MRS4CD-Algorithmus, der iterative FMLE-Ansatz oder die automatisierte Schwellenwertbestimmung. Damit soll vermieden werden, dass der Analyst sich in der Methode des Versuch und Irrtum verliert. Trotz dieser Hilfestellungen ist die Interpretation der Ergebnisse der objekt-basierten Änderungsdetektion immer noch Aufgabe des menschlichen Analysten. Ein teilweise automatisierter Prozess soll den Analysten daher nicht ersetzen, sondern unterstützen.

## 4. Zusammenfassung und Ausblick

### 4.1. Zusammenfassung

Die vorliegende Arbeit stellte eine umfassende Betrachtung objekt-basierter Verfahren zur Änderungsdetektion an. Außerdem wurde eine neuartige Methode zur vollständig objekt-basierten Änderungsdetektion entwickelt, implementiert und sowohl anhand simulierter als auch realer Daten getestet. Im Besonderen sind die folgenden Aspekte der Arbeit herauszustellen.

Für die **Segmentierung** im Rahmen der objekt-basierten Änderungsdetektion konnte ein Algorithmus, der MRS4CD-Algorithmus, entwickelt werden, der die beiden zu untersuchenden Aufnahmen einerseits adaptiv segmentiert und andererseits nur in solchen Bildbereichen Objekte unterschiedlich extrahiert, in denen auch relevante Unterschiede zwischen den Bildern festzustellen sind. Zur Umsetzung dieses Verfahrens wurden im Vergleich zu Listner (2008) eine weitere Methode zur Konsistenzüberprüfung, der *Relative Threshold Test*, und die Bereinigungsmethode *Local Removal Strategy* entwickelt. Der Segmentierungsalgorithmus wurde als Plugin für *eCognition* implementiert und dabei in Speicherbedarf und Laufzeitverhalten optimiert. Auch wurde der Rechenaufwand und Speicherbedarf sowohl in der Theorie als auch durch Messungen in der Praxis ermittelt. Im Vergleich zur Implementierung aus Listner (2008), bei der wegen der langen Laufzeit nur ein prototypischer Einsatz möglich ist, bietet das im Rahmen dieser Arbeit entwickelte Plugin eine praxistaugliche Lösung. An die Schnelligkeit des MRS-Algorithmus in *eCognition* reicht jedoch auch diese Implementierung nicht heran.

Darüber hinaus konnte im Rahmen dieser Arbeit ein Segmentierungsansatz für große Einzelbilder auf Grundlage des *Tiling-Stitching*-Verfahrens entwickelt und implementiert werden. Dieses Verfahren beruht auf dem MRS-Algorithmus. Jedoch ermöglicht der hier vorgestellte Ansatz im Gegensatz zum Original, dass große Bilder in Teilbilder zerlegt, parallel verarbeitet und anschließend die Segmentationen wieder nahtlos zusammengeführt werden können. Auf diese Weise wird der Speicherbedarf für die Segmentierung dieser Bilder erheblich reduziert. Im Hinblick auf die **Merkmalsextraktion** wurden in dieser Arbeit einerseits zwei Methoden zur Verknüpfung von Segmentationen vorgestellt. Die erste Methode, *Directed Change Detection* (DCD) genannt, ordnet jedem Objekt in der Segmentation des ersten Aufnahmezeitpunktes all jene Objekte in der Segmentation des zweiten Zeitpunktes zu, mit denen es eine räumliche Überlappung gibt. Das zweite Verfahren, die *Change Detection Using Intersected Objects* (CDIO), erzeugt räumliche Schnittmengen zwischen den Objekten des ersten und zweiten Aufnahmezeitpunkts und erreicht dadurch eine eindeutige Zuordnung zwischen den Segmentationen. Andererseits konnte ein Verfahren zur Merkmalsauswahl entwickelt werden. Diese Methode selektiert anhand der Entropie jene Objekteigenschaften für die weitere Verarbeitung, die den höchsten Informationsgehalt besitzen. Dadurch entfällt die aufwändige manuelle Auswahl der Objekteigenschaften.

Die **Transformation** des objekt-basierten Merkmalsraumes zur Betonung von relevanten Änderungen wurde im Rahmen dieser Arbeit mithilfe zweier Verfahren der multivariaten Statistik umgesetzt, die beide als Plugins für *eCognition* implementiert werden bzw. die Implementierung aus John (2009) angepasst werden konnten. Das erste der beiden Verfahren, der IR-MAD-Algorithmus, betrachtet die beiden Bilder mit ihren Objekteigenschaften als zwei Datensätze in einem Merkmalsraum. Das Verfahren hebt durch die Drehung der Daten-

sätze die Unterschiede zwischen den Datensätzen hervor und ermöglicht auf diese Weise eine Analyse der Unterschiede. Allerdings treten bei der Erweiterung des IR-MAD-Algorithmus von pixel- auf objekt-basierte Datensätze numerische Probleme auf. Diese Probleme können durch Anwendung eines zweiten Algorithmus der multivariaten Statistik, der PCA-Methode, behoben werden. Dieser Algorithmus zeichnet sich dadurch aus, dass die Gesamtvariabilität des Datensatzes durch eine reduzierte Anzahl Objekteigenschaften repräsentiert werden kann. Da diese begrenzte Menge von Variablen gleichzeitig untereinander unkorreliert ist, können die numerischen Probleme des IR-MAD-Algorithmus bei der objekt-basierten Änderungsdetektion gelöst werden, indem die PCA-Methode vorgeschaltet wird.

Getrennt von der IR-MAD-Methode wurde der PCA-Algorithmus im Rahmen dieser Arbeit auch als iterative Methode genutzt, um Unterschiede zwischen den beiden Aufnahmezeitpunkten zu detektieren. Dieser Ansatz kann jedoch im Gegensatz zur IR-MAD-Methode nur die Daten jeweils eines Objektmerkmals der beiden Aufnahmezeitpunkte für die Analyse verwenden.

Im Bereich der **Klassifikation** kam in dieser Arbeit auf der einen Seite das unüberwachte FMLE-Verfahren zum Einsatz. Dieser Algorithmus ordnet die Objekte auf Grundlage des transformierten Merkmalsraumes einer von mehreren Änderungskategorien zu. Das Verfahren muss dabei hinsichtlich der Anzahl der zur Verfügung stehenden Kategorien parametrisiert werden, wobei die vorliegende Arbeit eine Prozedur zur Wahl dieses Parameters vorschlagen konnte. Darüber hinaus wurde der FMLE-Algorithmus zu einem halbautomatischen, iterativen Verfahren erweitert, das unveränderte Objekte von der weiteren Analyse ausschließt.

Auf der anderen Seite wurde das schwellenwert-basierte Klassifikationsverfahren SEaTH zur Trennung von veränderten und unveränderten Objekten anhand des Z-Wertes der IR-MAD-Transformation benutzt. Diese Methode konnte im Rahmen dieser Arbeit von der Normalverteilung auf allgemeine Wahrscheinlichkeitsverteilungen erweitert werden. Dadurch ist eine stichprobenbasierte Auswahl des optimalen Schwellenwertes möglich. Das Verfahren wurde im Rahmen dieser Arbeit als Skript für das Statistikprogramm *R* implementiert.

Für die **Nachverarbeitung** der Ergebnisse der Änderungsdetektion wurden in dieser Arbeit der Douglas-Peucker-Algorithmus eingesetzt, um die Datenmenge zur Repräsentation der Objekte zu reduzieren und gleichzeitig eine Kurvenglättung durchzuführen.

Um die Güte der Ergebnisse zu bewerten, wurden Verfahren der Genauigkeitsuntersuchung betrachtet. Methoden zur Untersuchung der Qualität sowohl der Segmentierung als auch der Änderungsklassifikation wurden dabei vorgestellt.

Die genannten Teilverfahren wurden im Rahmen dieser Arbeit in einen Gesamtprozess integriert und in *eCognition* über die entsprechenden Plugins bzw. unter Zuhilfenahme von Regelsätzen implementiert. Anschließend wurde das Gesamtverfahren in Fallstudien mit unterschiedlichen Schwerpunkten auf simulierte als auch reale Daten angewendet. Die Ergebnisse wurden diskutiert und Schlussfolgerungen gezogen.

Insgesamt konnte gezeigt werden, dass das im Rahmen dieser Arbeit entwickelte Verfahren zur objekt-basierten Änderungsdetektion in praktischen Anwendungen umsetzbar ist und gute Ergebnisse liefern kann. Allerdings hat sich gezeigt, dass die vollständig objekt-basierte Änderungsdetektion ein sehr komplexes Verfahren darstellt. Dass der zusätzliche Aufwand auch zu einer Verbesserung der Ergebnisqualität führt, konnte leider nicht gezeigt werden. Auch ist anzumerken, dass für das Verfahren eine Vielzahl von Parametern bestimmt werden müssen. Die Versuche, diese Parameter automatisch zu bestimmen, verbessern die Qualität des Ergebnisses nicht in jedem Fall. Hilfestellungen für den Anwender oder halbautomatisierte Verfahren sind daher einer vollständig automatisierten Lösung vorzuziehen.

## 4.2. Ausblick

Als Konsequenz aus den Entwicklungen und Betrachtungen dieser Arbeit sind für zukünftige Forschungen einige Schlussfolgerungen zu ziehen.

Hinsichtlich der Vergleichbarkeit von Methoden zur Segmentierung wäre eine Datenbank mit Referenzsegmentationen wünschenswert, wie sie in Haindl und Mikes (2008) für die Textursegmentierung zu finden ist. In diesem Zusammenhang wäre auch die Nutzung von Bildsimulationstechniken zur Generierung künstlicher Bilder denkbar, damit gewisse Störeinflüsse in den Daten gezielt untersucht werden können. Die Weiterentwicklung von Methoden zur Betrachtung der Segmentierungsgüte ist dafür ebenfalls unerlässlich.

Der Segmentierungsalgorithmus MRS4CD selbst sollte im Hinblick auf zwei Aspekte weitere Entwicklung erfahren. Erstens ist der Algorithmus so zu erweitern, dass die in Abschnitt 2.3.5 vorgestellte *Tiling-Stitching*-Methode auch für die bitemporale Segmentierung eingesetzt werden kann. Zweitens ist mit Hinblick auf die Nutzung hierarchiebasierter Eigenschaften auch für den MRS4CD-Algorithmus die konsequente Umsetzung des Konzepts Multiskaligkeit nötig, wie dies im Falle der Einzelbildsegmentierung mit dem MRS-Algorithmus bereits realisiert wurde.

Obwohl der Ansatz der Merkmalsraumtransformation mit PCA und IR-MAD gute Ergebnisse zeigt, sind für die Zukunft auch die Benutzung anderer Methoden wie die Transformation *Maximum Noise Fraction* (siehe Canty 2009) für den Fall der objekt-basierten Änderung vorstellbar. Im Hinblick auf die Erweiterung der objekt-basierten Änderungsdetektion auf mehr als zwei Bilder sind zudem Methoden wie die *Multi-view Canonical Correlation Analysis* (siehe Nielsen 2002) interessant. Der Einsatz von Kernel-Methoden bei der objekt-basierten Änderungsdetektion zur Erweiterung der linearen Transformationen um nicht-lineare Effekte scheint ebenfalls sinnvoll (siehe Canty 2009).

Bei den Klassifikationsverfahren sollten weitere Entwicklungen in der objekt-basierten Bildanalyse unter Einbeziehung von Techniken des maschinellen Lernens erfolgen. Beispielsweise könnten andere unüberwachte Klassifikationsmethoden für *eCognition* bereitgestellt werden oder der FMLE-Algorithmus um andere Wahrscheinlichkeitsverteilungen erweitert werden.

Die im Rahmen dieser Arbeit vorgestellte Methode der objekt-basierten Änderungsdetektion arbeitet auf nur einer Maßstabsebene. Eine Erweiterung auf die gleichzeitige Verarbeitung mehrerer Ebenen, wie es in Schiewe (2002) vorgeschlagen wurde, erscheint daher sinnvoll.

In Bezug auf die Perspektiven von GEOBIA allgemein ist zu sagen, dass zwar einerseits neue Konzepte wie die Nutzung von Ontologien (siehe z.B. Arvor u. a. 2013), sowie die große Nutzergruppe, steigende Rechnerkapazitäten und weitere Anwendungsbereiche eine große Zukunft dieses Paradigmas versprechen. Andererseits existieren aber auch Gefahren wie die noch nicht abgeschlossene Operationalisierung des Prozesses, teilweise unklare Definitionen und daraus entstehende Missverständnisse und Verwirrung sowie eine Fragmentierung des Anwendungsfeldes statt einer Konsolidierung.

Ob sich das Konzept der GEOBIA innerhalb der Fernerkundung generell durchsetzen wird, muss somit die Zukunft zeigen. Eine Weiterentwicklung wird dieser Forschungsbereich in jedem Fall erfahren.

Für die Verifikationsarbeit der IAEO wird die Bedeutung von Fernerkundungsdaten in der Zukunft voraussichtlich weiter steigen, da einerseits die Anzahl der zu überwachenden nuklearen Anlagen wächst und andererseits das Budget der Organisation stagniert. Die Nutzung von Mitteln der Fernerkundung, um die Abläufe der Inspektionen zu optimieren oder sogar die Korrektheit und Vollständigkeit der Deklarationen anhand der Satellitenbilder zu verifizieren, erscheint daher sinnvoll. Um diesen Herausforderungen zu begegnen, sollten jedoch auch die Abläufe zur Auswertung von Fernerkundungsdaten innerhalb der IAEO angepasst werden. Insbesondere durch das steigende Datenvolumen scheint ein höherer Automatisierungsgrad dieser Prozesse angebracht. Verfahren zur automatisierten Änderungsdetektion können dazu



einen Beitrag leisten.

Allerdings sind aus Sicht des Autors beim Einsatz dieser Methoden drei Voraussetzungen zu schaffen. Erstens sollte eine nahtlose Integration der Nutzung automatisierter Analysewerkzeuge in die Geschäftsprozesse der IAEO gegeben sein. Zweitens ist die Schulung der Analysten essentiell und die dafür benötigten Ressourcen sind zu bedenken. Drittens sollte kein blindes Vertrauen in automatisierte Verfahren zur Änderungsdetektion gesetzt werden. Ihr Zweck sollte nämlich nicht sein, den Analysten zu ersetzen, sondern ihn bestmöglich zu unterstützen.

## A. Simulation von Bildpaaren

Um Methoden der Änderungsdetektion zu testen, werden in vielen Arbeiten simulierte Änderungen herangezogen. Auch in dieser Arbeit wird dieses Vorgehen in Kapitel 2 gewählt, um die Funktionsweise und Wirkung der vorgestellten Methoden zu demonstrieren.

Für die vorliegende Arbeit wurde ausgehend von einem Bild, das den ersten Aufnahmezeitpunkt repräsentiert, ein zweites Bild mit simulierten Änderungen erzeugt. Die Änderungen betreffen einerseits ein Haus, das aus dem zentralen, oberen Bildbereich kopiert und in die linke untere Ecke des Bildes eingefügt wurde. Andererseits wurde mithilfe der Funktion `noise` aus der *DIPImage*-Toolbox für *Matlab* (siehe Hendriks u. a. 2012) Rauschen hinzugefügt.

Formal wird für ein Bild  $I$  das zugehörige simulierte Änderungsbild  $I'$  erzeugt, indem eine Zufallsgröße  $B$  addiert wird, d.h.

$$I' = I + B. \quad (\text{A.1})$$

Für die Einträge der Matrix  $B = (b_{i,j})$  gilt dabei

$$b_{i,j} \sim \mathcal{N}(0; \sigma^2). \quad (\text{A.2})$$

Dabei sind die  $b_{i,j}$  unabhängig und identisch verteilt.

Dies entspricht additivem Gaußschem Rauschen mit Mittelwert 0 und verschiedenen Werten für die Standardabweichung  $\sigma$ . Der Wertebereich für  $I$  und  $I'$  ist hierbei auf natürliche Zahlen zwischen 0 und 255 beschränkt und Werte für  $I'$  außerhalb dieser Menge werden gerundet bzw. auf den Rand abgebildet.

Über den ersten Parameter der Funktion `noise` kann die Verteilung von  $B$  z.B. auf eine Gleich- oder Poissonverteilung angepasst werden. Darüber hinaus ist durch die Arithmetik der *DIPImage*-Toolbox die Möglichkeit gegeben, dem Bild multiplikatives Rauschen durch eine zweite Zufallsgröße  $A$  hinzuzufügen, d.h.

$$I' = AI + B. \quad (\text{A.3})$$

Im Rahmen dieser Arbeit wurde die Bildsimulation jedoch auf additives Gaußsches Rauschen und das Verschieben eines Objektes per Copy-Paste beschränkt.

Der Ausgangspunkt und Ergebnis der Bildsimulation sind in Abbildung A.1 beispielhaft dargestellt. Auf der dieser Arbeit beiliegenden CD findet sich das *Matlab*-Skript zur Generierung des Bildes sowie weitere Beispielergebnisse im Verzeichnis `ChangeSimulation`.



(a) Ursprungsbild.



(b) Ergebnis.

**Abb. A.1.:** Beispielergebnis für die Bildsimulation im Rahmen der objekt-basierten Änderungsdetektion ( $\sigma = 50$ ).

# Literatur

- Abam, M. u. a. (2010). „Streaming Algorithms for Line Simplification“. In: *Discrete & Computational Geometry* 43 (3), S. 497–515.
- Abmayr, W. (1994). *Einführung in die digitale Bildverarbeitung*. Stuttgart: Teubner.
- Albertz, Jörg (2009). *Einführung in die Fernerkundung: Grundlagen der Interpretation von Luft- und Satellitenbildern*. 4. Aufl. WBG.
- Albrecht, F. u. a. (2010). „Spatial accuracy assessment of object boundaries for object-based image analysis“. In: *Proceedings of the GEOBIA 2010 Conference*.
- Aplin, P. (2004). „Remote sensing: land cover“. In: *Progress in physical geography* 28.2, S. 283–293.
- Arouja Sousa, M. (2009). *SegSAR - Image Segmentation*. Online verfügbar unter <https://sites.google.com/site/segsar2/program> (Abrufdatum 01.07.2012).
- Arvor, Damien u. a. (2013). „Advances in geographic object-based image analysis with ontologies: a review of main contributions and limitations from a remote sensing perspective“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 82, S. 125–137.
- Asano, T. (2001). „Efficient algorithms for optimization-based image segmentation“. In: *International Journal of Computational Geometry & Applications*.
- Aubert, G. und P. Kornprobst (2006). *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Bd. 147. Springer.
- Aytekin, Örsan und Ilkay Ulusoy (2011). „Automatic segmentation of VHR images using type information of local structures acquired by mathematical morphology“. In: *Pattern Recognition Letters* 32.13, S. 1618–1625.
- Baatz, D. u. a. (2003). *Digital image segmentation method*. EP Patent 0,858,051.
- Baatz, M. und A. Schäpe (2000). „Multiresolution Segmentation - an Optimization Approach for High Quality Multi-Scale Image Segmentation“. In: *Angewandte Geographische Informationsverarbeitung XII. Beiträge zum AGIT-Symposium Salzburg 2000*. Wichmann, S. 12–23.
- Ballard, D. H. und C. M. Brown (1982). *Computer Vision*. Prentice-Hall.
- Baute, J. (Apr. 2011). *Iraq Case Study: 14 Years of Ongoing Challenges (April 1991 - July 2005)*. ESARDA course on Nuclear Safeguards and Non-Proliferation.
- Beauchemin, M. u. a. (1998). „On the Hausdorff distance used for the evaluation of segmentation results“. In: *Canadian journal of remote sensing* 24.1, S. 3–8.
- Benz, U. C. u. a. (2004). „Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 58.3–4, S. 239–258.
- Biehl, L. und D. Landgrebe (2002). „MultiSpec—a tool for multispectral-hyperspectral image data analysis“. In: *Computers & Geosciences* 28.10, S. 1153–1159.
- Bins, L. S. u. a. (1996). „Satellite imagery segmentation: a region growing approach“. In: *Proceedings of VIII Brazilian Remote Sensing Symposium*, S. 33–62.
- Blaschke, T. (2005). „A framework for change detection based on image objects“. In: *Remote Sensing & GIS for Environmental Studies: Applications in Geography*. Hrsg. von Stefan Erasmí u. a., S. 1–9.
- (2010). „Object based image analysis for remote sensing“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 65.1, S. 2–16.

- Blaschke, T. u. a., Hrsg. (2008). *Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications*. Springer.
- Bong, C. W. und M. Rajeswari (2012). „Multiobjective clustering with metaheuristic: current trends and methods in image segmentation“. In: *IET Image Processing* 6.1, S. 1–10.
- Bouman, C. A. und M. Shapiro (1994). „A multiscale random field model for Bayesian image segmentation“. In: *IEEE Transactions on Image Processing* 3.2, S. 162–177.
- Bronstein, I. N. u. a. (2005). *Taschenbuch der Mathematik*. 6. Aufl. Harri Deutsch Verlag.
- Brückmann, M. (2004). „Farbbildsegmentierung mit Fuzzy-Methoden“. Magisterarb. Technische Universität Ilmenau.
- Bruzzone, L. und D. F. Prieto (2000). „Automatic Analysis of the Difference Image for Unsupervised Change Detection“. In: *IEEE Transactions on Geoscience and Remote Sensing, ITGRS* 38.3, S. 1171–1182.
- (2002). „An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images“. In: *IEEE Transactions on Image Processing, ITIP* 11.4, S. 452–466.
- Burnett, C. und T. Blaschke (2003). „A multi-scale segmentation/object relationship modelling methodology for landscape analysis“. In: *Ecological modelling* 168.3, S. 233–249.
- Butenuth, M. und C. Heipke (2012). „Network snakes: graph-based object delineation with active contour models“. In: *Machine Vision and Applications* 23.1, S. 91–109.
- Camps-Valls, G. und L. Bruzzone (Juni 2005). „Kernel-based methods for hyperspectral image classification“. In: *Geoscience and Remote Sensing, IEEE Transactions on* 43.6, S. 1351 – 1362.
- Canty, M. J. (2009). *Image Analysis, Classification, And Change Detection In Remote Sensing: With Algorithms For ENVI/IDL*. 2nd. Taylor & Francis.
- Canty, M. J. und A. A. Nielsen (März 2008). „Automatic Radiometric Normalization of Multitemporal Satellite Imagery with the Iteratively Re-weighted MAD Transformation“. In: *Remote Sensing of Environment* 112.3, S. 1025–1036.
- Canty, M. J. u. a. (2004). „Automatic radiometric normalization of multitemporal satellite imagery“. In: *Remote Sensing of Environment* 91.3-4, S. 441–451.
- Carleer, A. P. u. a. (2005). „Assessment of very high spatial resolution satellite image segmentations“. In: *Photogrammetric Engineering and Remote Sensing* 71.11, S. 1285–1294.
- Castilla, G. u. a. (2008). „Size-constrained region merging (SCRM): an automated delineation tool for assisted photointerpretation“. In: *Photogrammetric Engineering and Remote Sensing* 74.4, S. 409.
- Cesar, R. M. u. a. (2005). „Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms“. In: *Pattern Recognition* 38.11, S. 2099–2113.
- Chan, T. F. u. a. (1979). *Updating Formulae and a Pairwise Algorithm for Computing Sample Variances*. Techn. Ber. DTIC Document.
- Chen, B. u. a. (2010). „Algorithm for aerial image segmentation based on Mean Shift with wavelet“. In: *Jisuanji Gongcheng yu Yingyong (Computer Engineering and Applications)* 46.12, S. 143–145.
- Chen, G. u. a. (2008). „A Comparison of Object-Based and Pixel-Based Approaches to Estimate Forest Canopy Height Using QuickBird Imagery“. In: *Proceedings of the GEOBIA 2008 Conference: Pixels Objects Intelligence Geographic Object Based Image Analysis for the 21st Century*.
- Chen, G. u. a. (2012). „Object-based change detection“. In: *International Journal of Remote Sensing* 33.14, S. 4434–4457.
- Chen, Z. Q., T. C. Hutchinson u. a. (2007). „Urban damage estimation using statistical processing of satellite images“. In: *Journal of computing in civil engineering* 21, S. 187.
- Cheng, H. D. u. a. (2001). „Color image segmentation: advances and prospects“. In: *Pattern Recognition* 34.12, S. 2259–2281.

- Christoudias, C. M. u. a. (2002). „Synergism in low level vision“. In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Bd. 4. IEEE, S. 150–155.
- Clinton, N. u. a. (2010). „Accuracy assessment measures for object-based image segmentation goodness“. In: *Photogrammetric Engineering and Remote Sensing* 76.3, S. 289–299.
- Comaniciu, D. und P. Meer (2002). „Mean shift: A robust approach toward feature space analysis“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, ITPAMI* 24.5, S. 603–619.
- Conchedda, G. u. a. (2008). „An object-based method for mapping and change analysis in mangrove ecosystems“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 63.5, S. 578–589.
- Copernicus - The European Earth Observation Programme (2013). Englisch. Online verfügbar unter <http://www.copernicus.eu> (Abrufdatum 24.08.2013).
- Coppin, P. R. und M. E. Bauer (1996). „Digital change detection in forest ecosystems with remote sensing imagery“. In: *Remote sensing reviews* 13.3-4, S. 207–234.
- Coppin, P. u. a. (2004). „Digital change detection methods in ecosystem monitoring: a review“. In: *International Journal of Remote Sensing* 25, S. 1565–1596.
- Cplusplus.com. *function template std::random\_shuffle*. Online verfügbar unter [http://www.cplusplus.com/reference/algorithm/random\\_shuffle/](http://www.cplusplus.com/reference/algorithm/random_shuffle/) (Abrufdatum 04.10.2012).
- Cross, A. M. u. a. (1988). „Segmentation of remotely-sensed images by a split-and-merge process“. In: *International Journal of Remote Sensing* 9.8, S. 1329–1345.
- Cushnie, J. L. (1987). „The interactive effect of spatial resolution and degree of internal variability within land-cover types on classification accuracies“. In: *International Journal of Remote Sensing* 8.1, S. 15–29.
- D’Agostino, R. B. und M. A. Stephens (1986). *Goodness-Of-Fit Techniques*. Marcel Dekker, Inc.
- D’Elia, C. u. a. (2003). „A tree-structured Markov random field model for Bayesian image segmentation“. In: *IEEE Transactions on Image Processing, ITIP* 12.10, S. 1259–1273.
- Daneke, C. (2008). „Automatisierte Analyse von Fernerkundungsdaten für umfangreiche Monitoring-Aufgaben im Rahmen nuklearer Safeguards“. Masterarbeit. Universität Wien.
- De Kok, R. (2001). „Objektorientierte Bildanalyse“. Diss. Technische Universität München, Universitätsbibliothek.
- Definiens Imaging, Munich (2009). *Definiens eCognition Developer 8 User Guide*.
- Dekker, R. u. a. (2009). „Change Detection Tools“. In: *Remote Sensing from Space: Supporting International Peace and Security*. Hrsg. von B. Jasani. Springer, S. 119–140.
- Delafontaine, M. u. a. (2009). „Assessment of sliver polygons in geographical vector data“. In: *International Journal of Geographical Information Science* 23.6, S. 719–735.
- Dempster, A. P. u. a. (1977). „Maximum likelihood from incomplete data via the EM algorithm“. In: *Journal of the Royal Statistical Society. Series B (Methodological)*, S. 1–38.
- Desclée, B. u. a. (2006). „Tropical forest monitoring by object-based change detection: towards an automated method in an operational perspective“. In: *Proceedings of the OBIA 2006 Conference - 1st International Conference on Object-based Image Analysis*, S. 4–5.
- Dey, V. u. a. (2010). „A review on image segmentation techniques with remote sensing perspective“. In: *Proceedings of the International Society for Photogrammetry and Remote Sensing Symposium (ISPRS10)*. Bd. 38, S. 5–7.
- Donoser, M. u. a. (2009). „Saliency driven total variation segmentation“. In: *Proceedings of the 12th IEEE International Conference on Computer Vision, ICCV*. IEEE, S. 817–824.
- Donoser, M. u. a. (2011). „Highly consistent sequential segmentation“. In: *Image Analysis*, S. 48–58.
- Douglas, D. H. und T. K. Peucker (1973). „Algorithms for the reduction of the number of points required to represent a digitized line or its caricature“. In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 10.2, S. 112–122.

- Draguta, L. u. a. (2010). „ESP: a tool to estimate scale parameter for multiresolution image segmentation of remotely sensed data“. In: *International Journal of Geographical Information Science* 24.6, S. 859–871.
- Duda, R. O. u. a. (2001). *Pattern classification*. Wiley.
- ERDAS Imagine (2012). *Feature Extraction & Classification*. Online verfügbar unter [http://www.erdas.com/Libraries/White\\_Papers/Feature\\_Extraction\\_and\\_Classification.sflb.ashx](http://www.erdas.com/Libraries/White_Papers/Feature_Extraction_and_Classification.sflb.ashx) (Abrufdatum 26.04.2012). ERDAS Inc.
- Ebisch, K. (2002). „A correction to the Douglas-Peucker line generalization algorithm“. In: *Computers & geosciences* 28.8, S. 995–997.
- Esch, T. u. a. (Juli 2008). „Improvement of Image Segmentation Accuracy Based on Multiscale Optimization Procedure“. In: *IEEE Geoscience and Remote Sensing Letters* 5.3, S. 463 – 467.
- Feitosa, R. Q. u. a. (2006). „A genetic approach for the automatic adaptation of segmentation parameters“. In: *Proceedings of the OBIA 2006 Conference - 1st International Conference on Object-based Image Analysis*. Bd. 4.
- Felzenszwalb, P. F. und D. P. Huttenlocher (2004). „Efficient graph-based image segmentation“. In: *International Journal of Computer Vision* 59.2, S. 167–181.
- Feng, W. u. a. (Okt. 2010). „Self-Validated Labeling of Markov Random Fields for Image Segmentation“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, ITPAMI* 32.10, S. 1871–1887.
- Ferro-Famil, L. u. a. (2003). „Statistical segmentation of polarimetric SAR data“. In: *Proceedings of the Workshop on POLinSAR-Applications of SAR Polarimetry and Polarimetric Interferometry*. Hrsg. von H. Lacoste. CD-ROM., p. 15.1.
- Friedland, G. u. a. (2005). „SIOX: simple interactive object extraction in still images“. In: *Proceedings of the Seventh IEEE International Symposium on Multimedia*. IEEE, 7 pp.
- Galun, M. u. a. (2003). „Texture segmentation by multiscale aggregation of filter responses and shape elements“. In: *Proceedings of the Ninth International Conference on Computer Vision, ICCV*, S. 716–723.
- Gamanya, R. u. a. (2009). „Object-oriented change detection for the city of Harare, Zimbabwe“. In: *Expert Systems with Applications* 36.1, S. 571–588.
- Gao, Y. und J. F. Mas (2008). „A comparison of the performance of pixel-based and object-based classifications over images with various spatial resolutions“. In: *Online Journal of Earth Sciences* 2.1, S. 27–35.
- Gath, I. und A. B. Geva (Juli 1989). „Unsupervised optimal fuzzy clustering“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, ITPAMI* 11.7, S. 773–780.
- Gofman, E. (2006). „Developing an Efficient Region Growing Engine for Image Segmentation“. In: *IEEE International Conference on Image Processing*, S. 2413–2416.
- Gonçalves, J. G. M. u. a. (2010). „Integrated Analysis of Satellite Imagery for Treaty Monitoring - The LIMES Experience“. In: *ESARDA Bulletin* 43.
- González, A. u. a. (2009). „Nonparametric segmentation of clouds from multispectral MSG-SEVIRI imagery“. In: *Proceedings of SPIE*. Bd. 7475, S. 747513.
- Grazzini, J. u. a. (2010). „Simultaneous hierarchical segmentation and vectorization of satellite images through combined non-uniform data sampling and anisotropic triangulation“. In: *Proceedings of SPIE*. Bd. 7830, 78300F.
- Hahmann, S. (2006). „Automatische Generalisierungsverfahren zur Vereinfachung von Kartenvektordaten unter Berücksichtigung der Topologie und Echtzeitfähigkeit“. Masterarbeit. Hochschule für Technik und Wirtschaft (HTW) Dresden.
- Haindl, M. und S. Mikes (2008). „Texture segmentation benchmark“. In: *Proceedings of the 19th International Conference on Pattern Recognition, ICPR*.
- Hall-Beyer, M. (2007). *The GLCM Tutorial Home Page*. Online verfügbar unter <http://www.fp.ucalgary.ca/mhallbey/tutorial.htm> (Abrufdatum 28.06.2011).

- Happ, P. N. u. a. (2010). „Multiresolution Segmentation: A parallel approach for high resolution image segmentation in multicore architectures“. In: *Proceedings of the GEOBIA 2010 Conference*.
- Haralick, R. M. und L. G. Shapiro (1985). „Image segmentation techniques“. In: *Computer vision, graphics, and image processing* 29.1, S. 100–132.
- (1992). *Computer and Robot Vision*. Addison-Wesley.
- Haralick, R. M. u. a. (1973). „Textural features for image classification“. In: *IEEE Transactions on Systems, Man and Cybernetics* 3.6, S. 610–621.
- Hay, G. J. und G. Castilla (2006). „Object-based image analysis: strengths, weaknesses, opportunities and threats (SWOT)“. In: *Proceedings of the OBIA 2006 Conference - 1st International Conference on Object-based Image Analysis*.
- (2008). „Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline“. In: *Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications*. Hrsg. von T. Blaschke u. a. Springer, S. 75–89.
- Heckbert, P. S. und M. Garland (1997). *Survey of polygonal surface simplification algorithms*. Techn. Ber. DTIC Document.
- Hendriks, C. L. L. u. a. (2012). *DIPimage User Manual*. Online verfügbar unter [ftp://ftp.tudelft.nl/pub/DIPimage/latest/docs/dipimage\\_user\\_manual.pdf](ftp://ftp.tudelft.nl/pub/DIPimage/latest/docs/dipimage_user_manual.pdf), (Abrufdatum 12.02.2013).
- Henry, C. und J. F. Peters (2008). „Near set index in an objective image segmentation evaluation framework“. In: *Proceedings of the GEOBIA 2008 Conference: Pixels Objects Intelligence Geographic Object Based Image Analysis for the 21st Century*, S. 1–8.
- Herold, H. (2005). „Beiträge zum Vergleich von Segmentierungsprogrammen fernerkundlicher Bilddaten“. Masterarbeit. Institut für Geographie der TU Dresden.
- Hershberger, J. und J. Snoeyink (1992). „Speeding Up the Douglas-Peucker Line-Simplification Algorithm“. In: *Proceedings of the 5th International Symposium on Spatial Data Handling*, S. 134–143.
- Hese, S. und C. Schmullius (2006). „Object context information for advanced forest change classification strategies“. In: *Proceedings of the OBIA 2006 Conference - 1st International Conference on Object-based Image Analysis*, S. 04–05.
- Hettling, J. K. (2008). *Satellite Imagery for Verification and Enforcement of Public International Law*. Schriften zum Luft- und Weltraumrecht. Carl Heymanns.
- Hofmann, P. u. a. (2008). „Detecting informal settlements from QuickBird data in Rio de Janeiro using an object based approach“. In: *Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications*. Hrsg. von T. Blaschke u. a. Springer, S. 531–553.
- Horn, R. A. und C. R. Johnson (1999). *Matrix Analysis*. Cambridge University Press.
- Howe, D. (1995). *The free on-line dictionary of computing*. Online verfügbar unter <http://foldoc.org> (Abrufdatum 10.04.2012).
- Im, J. u. a. (2008). „Object-based change detection using correlation image analysis and image segmentation“. In: *International Journal of Remote Sensing* 29.2, S. 399–423.
- Institute of Scientific Information (ISI) (2012). *Web of Science*. Online verfügbar unter <http://www.webofscience.com> (Abrufdatum 12.04.2012).
- Instituto Nacional de Pesquisas Espaciais (INPE) (2012). *SPRING - Georeferenced Information Processing System*. Online verfügbar unter <http://www.dpi.inpe.br/spring/english/index.html> (Abrufdatum 27.04.2012).
- Internationale Atomenergie-Organisation (IAEO) (Juni 1972). *THE STRUCTURE AND CONTENT OF AGREEMENTS BETWEEN THE AGENCY AND STATES REQUIRED IN CONNECTION WITH THE TREATY ON THE NON-PROLIFERATION OF NUCLEAR WEAPONS*. Online verfügbar unter <http://www.iaea.org/Publications/Documents/Infcircs/0thers/infcirc153.pdf> (Abrufdatum 01.09.2013).



- Internationale Atomenergie-Organisation (IAEO) (Sep. 1997). *MODEL PROTOCOL ADDITIONAL TO THE AGREEMENT(S) BETWEEN STATE(S) AND THE INTERNATIONAL ATOMIC ENERGY AGENCY FOR THE APPLICATION OF SAFEGUARDS (INFCIRC/540)*. Online verfügbar unter <http://www.iaea.org/Publications/Documents/Infcircs/1997/infcirc540c.pdf> (Abrufdatum 01.09.2013).
- (2001). „IAEA Safeguards Glossary“. In: *International Nuclear Verification Series No. 3*.
  - (2011). *Safeguards Statement for 2010*. Online verfügbar unter <http://www.iaea.org/safeguards/documents/es2010.pdf>.
  - (2012). *Safeguards Statement for 2011*. Online verfügbar unter <http://www.iaea.org/safeguards/documents/es2011.pdf>.
  - (2013). *Safeguards Statement for 2012*. Online verfügbar unter [http://www.iaea.org/safeguards/statements-repository/Section\\_AB\\_SIR\\_2012.pdf](http://www.iaea.org/safeguards/statements-repository/Section_AB_SIR_2012.pdf).
- Janakiraman, T. N. u. a. (2007). „Image Segmentation Based on Minimal Spanning Tree and Cycles“. In: *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications*. Bd. 3.
- John, A. (2009). „Objekt-basierte Änderungsdetektion“. Magisterarb. TU Bergakademie Freiberg.
- Karantzas, K. G. und D. P. Argialas (2003). „Evaluation of selected edge detection techniques in remotely sensing images“. In: *Proceedings of SPIE*. Bd. 4885, S. 102.
- Kauth, R. J. und G. S. Thomas (1976). „The tasseled cap—a graphic description of the spectral-temporal development of agricultural crops as seen by Landsat“. In: *LARS Symposia*, S. 159.
- Kennedy, Robert E. u. a. (2007). „Trajectory-based change detection for automated characterization of forest disturbance dynamics“. In: *Remote Sensing of Environment* 110.3, S. 370–386.
- Kettig, R. L. und D. A. Landgrebe (1976). „Classification of multispectral image data by extraction and classification of homogeneous objects“. In: *IEEE Transactions on Geoscience Electronics, ITGE* 14.1, S. 19–26.
- Kim, M. und M. Madden (2006). „Determination of optimal scale parameters for alliance-level forest classification of multispectral IKONOS images“. In: *Proceedings of the OBIA 2006 Conference - 1st International Conference on Object-based Image Analysis*.
- Kolesnikov, A. und P. Fränti (2007). „Polygonal approximation of closed discrete curves“. In: *Pattern Recognition* 40.4, S. 1282–1293.
- Kriegler, F. J. u. a. (1969). „Preprocessing transformations and their effects on multispectral recognition“. In: *Proceedings of the 6th International Symposium on Remote Sensing of Environment*. Bd. 1, S. 97–131.
- Kristinsdóttir, B. (2008). „Implications of Invariant Moments for Texture Analysis, Segmentation and Classification“. Magisterarb. TU Bergakademie Freiberg.
- Kux, H. J. H. und E. H. G. Araújo (2008). „Object-based Image Analysis using QuickBird satellite images and GIS data, case study Belo Horizonte (Brazil)“. In: *Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications*. Hrsg. von T. Blaschke u. a. Springer, S. 571–588.
- Lang, S. und T. Blaschke (2006). „Bridging remote sensing and GIS - What are the main supportive pillars?“ In: *Proceedings of the OBIA 2006 Conference - 1st International Conference on Object-based Image Analysis*.
- Lang, S. u. a. (2009). „Combined object-based classification and manual interpretation-Synergies for a quantitative assessment of parcels and biotopes“. In: *Geocarto International* 24.2, S. 99–114.
- Lee, D. (1996). „Automation of Map Generalization: The Cutting-Edge Technology“. In: *ESRI White Paper Series*.

- Leedham, G. u. a. (2002). „Separating text and background in degraded document images-a comparison of global thresholding techniques for multi-stage thresholding“. In: *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition*, S. 244–249.
- Leedham, G. u. a. (2003). „Comparison of some thresholding algorithms for text/background segmentation in difficult document images“. In: *Proceedings of the Seventh International Conference on Document Analysis and Recognition, ICDAR*, S. 859–864.
- Lefebvre, A. u. a. (2008). „Object-oriented approach and texture analysis for change detection in very high resolution images“. In: *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS*. Bd. 4. IEEE, S. 663–666.
- Leu, J. G. und L. Chen (1988). „Polygonal approximation of 2-D shapes through boundary merging“. In: *Pattern Recognition Letters* 7.4, S. 231–238.
- Li, P. und X. Xiao (2007). „Multispectral image segmentation by a multichannel watershed-based approach“. In: *International Journal of Remote Sensing* 28.19, S. 4429–4452.
- Li, S. Z. (2009). *Markov random field modeling in image analysis*. 3. Advances in Pattern Recognition. Springer.
- Li, W. u. a. (Sep. 2011). „Robust airplane detection in satellite images“. In: *Proceedings of the 18th IEEE International Conference on Image Processing, ICIP*, S. 2821–2824.
- Lillesand, T. M. u. a. (2008). *Remote sensing and image interpretation*. Ed. 6. John Wiley & Sons Ltd.
- Linting, M. u. a. (2011). „Statistical Significance of the Contribution of Variables to the PCA solution: An Alternative Permutation Strategy“. In: *Psychometrika* 76.3, S. 440–460.
- Listner, C. (2008). „Bildsegmentierung für die objektbasierte Änderungsdetektion digitaler Satellitenbilder“. Magisterarb. TU Bergakademie Freiberg.
- Listner, C. und I. Niemeyer (2011). „Object-based Change Detection“. In: *Photogrammetrie-Fernerkundung-Geoinformation, PFG* 2011.4, S. 233–245.
- Lizarazo, I. und P. Elsner (2009). „Fuzzy segmentation for object-based image classification“. In: *International Journal of Remote Sensing* 30.6, S. 1643–1649.
- Lu, D. u. a. (2004). „Change detection techniques“. In: *International Journal of Remote Sensing* 25.12, S. 2365–2401.
- Lucieer, A. (2004). „Uncertainties in Segmentation and their Visualisation“. Diss. Utrecht University, International Institute for Geo-Information Science und Earth Observation (ITC).
- MVtec (2011). *HALCON Referenzhandbuch 9.0.4*. MVtec Software GmbH.
- Mardia, K. V. u. a. (1979). *Multivariate analysis (probability and mathematical statistics)*.
- Marpu, P. R. (2009). „Geographic Object-based Image Analysis“. Diss. Institute of Mine-Surveying und Geodesy, TU Bergakademie Freiberg.
- Marpu, P. R. u. a. (2010). „Enhanced evaluation of image segmentation results“. In: *Journal of Spatial Science* 55.1, S. 55–68.
- Martin, D. u. a. (2001). „A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics“. In: *Proceedings of the Eighth IEEE International Conference on Computer Vision, ICCV*. Bd. 2. IEEE, S. 416–423.
- Mas, J. F. (1999). „Monitoring land-cover changes: a comparison of change detection techniques“. In: *International Journal of Remote Sensing* 20.1, S. 139–152.
- Mathworks, The (2005). *Getting started with MATLAB 7*. URL: [http://www.mathworks.com/access/helpdesk/help/pdf\\_doc/matlab/getstart.pdf](http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/getstart.pdf).
- Matthew, M. W. u. a. (2002). „Atmospheric correction of spectral imagery: evaluation of the FLAASH algorithm with AVIRIS data“. In: *Applied Imagery Pattern Recognition Workshop, 2002. Proceedings. 31st*, S. 157–163.
- McCauley, J. D. und B. A. Engel (1995). „Comparison of scene segmentations: SMAP, ECHO, and maximum likelihood“. In: *Geoscience and Remote Sensing, IEEE Transactions on* 33.6, S. 1313–1316.

- McDermid, G. J. u. a. (2008). „Object-based approaches to change analysis and thematic map update: challenges and limitations“. In: *Canadian Journal of Remote Sensing* 34.5, S. 462–466.
- Meinel, G. und M. Neubert (2004). „A Comparison of Segmentation Programs for High Resolution RemoteSensing Data“. In: *Proceedings of the XX ISPRS Congress*. Bd. XXXV Part 4, S. 1097–1102.
- Meinel, G. u. a. (2001). „The potential use of very high resolution satellite data for urban areas—First experiences with IKONOS data, their classification and application in urban planning and environmental monitoring“. In: *Regensburger Geographische Schriften* 35, S. 196–205.
- Meyer, F. (1994). „Topographic distance and watershed lines“. In: *Signal processing* 38.1, S. 113–125.
- Meyer, F. und P. Maragos (1999). „Multiscale Morphological Segmentations Based on Watershed, Flooding, and Eikonal PDE“. In: *Scale-Space Theories in Computer Vision*. Hrsg. von M. Nielsen u. a. Bd. 1682. Lecture Notes in Computer Science, S. 351–362.
- Moyland, Suzanna van (1997). *The IAEA's Safeguards Programme '93+2'*. Verification Matters. VERTIC.
- Mumford, D. und J. Shah (1989). „Optimal approximations by piecewise smooth functions and associated variational problems“. In: *Communications on pure and applied mathematics* 42.5, S. 577–685.
- N. A. Software Limited (2010). *InfoPACK User Guide Version 2.2*.
- National Institutes of Health (NIH) (2003). *ImageJ Segmentation Plugins*. abrufbar unter <http://rsbweb.nih.gov/ij/plugins/index.html#segmentation> (Abrufdatum 27.04.2012).
- Neubert, M. u. a. (2008). „Assessing image segmentation quality-concepts, methods and application“. In: *Object-based image analysis*, S. 769–784.
- Neumann, K. und M. Selke (2001). „Elemente der Programmiersprache Java vorgestellt an einer Modifikation des Douglas-Peucker-Algorithmus“. In: *Mitteilungen des Bundesamtes für Kartographie und Geodäsie*.
- Nielsen, A. A. (2002). „Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data“. In: *Image Processing, IEEE Transactions on* 11.3, S. 293–305.
- (2007). „The regularized iteratively reweighted MAD method for change detection in multi- and hyperspectral data“. In: *IEEE Transactions on Image Processing, ITIP* 16.2, S. 463–478.
- Nielsen, A. A. u. a. (1998). „Multivariate Alteration Detection (MAD) and MAF processing in multispectral, bitemporal image data: New approaches to change detection studies“. In: *Remote Sensing of Environment* 64, S. 1–19.
- Niemeyer, I. (2001). *Satelliten und nukleare Kontrolle - Änderungsdetektion und objektorientierte, wissenschaftliche Klassifikation von Multispektralaufnahmen zur Unterstützung der nuklearen Verifikation*. Jülich: Forschungszentrum Jülich.
- Niemeyer, I. u. a. (2005). „Automation of change detection procedures for nuclear safeguards-related monitoring purposes“. In: *International Geoscience and Remote Sensing Symposium*. Bd. 3. Citeseer, S. 2133.
- Niemeyer, I. u. a. (2008). „Change detection using object features“. In: *Object-Based Image Analysis*, S. 185–201.
- Nischwitz, A. und P. Haberäcker (2004). *Masterkurs Computergrafik und Bildverarbeitung alles für Studium und Praxis; Bildverarbeitungswerkzeuge, Beispiel-Software und interaktive Vorlesungen online verfügbar*. 1. Aufl. zahlr. Ill., graph. Darst. Wiesbaden: Vieweg.
- Noma, A. u. a. (2012). „Interactive image segmentation by matching attributed relational graphs“. In: *Pattern Recognition* 45.3, S. 1159–1179.

- Nussbaum, S. und G. Menz (2008). *Object-based image analysis and treaty verification: new approaches in remote sensing-applied to nuclear facilities in Iran*. Springer.
- Nussbaum, S. u. a. (2006). „SEATH-a new tool for automated feature extraction in the context of object-based image analysis“. In: *Proceedings of the OBIA 2006 Conference - 1st International Conference on Object-based Image Analysis*. Bd. 36. 4, S. C42.
- O’Callaghan, R. J. und D. R. Bull (2005). „Combined morphological-spectral unsupervised image segmentation“. In: *Image Processing, IEEE Transactions on* 14.1, S. 49–62.
- O’Neil-Dunne, J. P. M. u. a. (2012). „An object-based system for LIDAR data fusion and feature extraction“. In: *Geocarto International*.
- O’Neil-Dunne, J. u. a. (Aug. 2009). „Object-based high-resolution land-cover mapping“. In: *Proceedings of the 17th International Conference on Geoinformatics*, S. 1 –6.
- Osserman, R. (1978). „The isoperimetric inequality“. In: *Bull. Amer. Math. Soc* 84.6, S. 1182–1238.
- PCI Geomatics (2000). *APEX version 7.0 user’s manual*.
- Parvati, K. u. a. (2009). „Image segmentation using gray-scale morphology and marker-controlled watershed transformation“. In: *Discrete Dynamics in Nature and Society* 2008.
- Perez, C. B. und G. Olague (2007). „Unsupervised Evolutionary Segmentation Algorithm Based on Texture Analysis“. In: *Lecture Notes in Computer Science* 4448. Hrsg. von M. Giacobini, S. 407–414.
- Perveen, M. F. u. a. (2008). „Evaluation of ASTER spectral bands for agricultural land cover mapping using pixel-based and object-based classification approaches“. In: *Proceedings of the GEOBIA 2008 Conference: Pixels Objects Intelligence Geographic Object Based Image Analysis for the 21st Century*.
- Pesaresi, M. und J. A. Benediktsson (2001). „A new approach for the morphological segmentation of high-resolution satellite imagery“. In: *IEEE Transactions on Geoscience and Remote Sensing, ITGRS* 39.2, S. 309–320.
- Petrovic, S. (2006). „A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters“. In: *Proceedings of the 11th Nordic Workshop of Secure IT Systems*. Citeseer, S. 53–64.
- Poggi, G. u. a. (Aug. 2005). „Supervised segmentation of remote sensing images based on a tree-structured MRF model“. In: *IEEE Transactions on Geoscience and Remote Sensing, ITGRS* 43.8, S. 1901–1911.
- Quantum GIS Development Team (2009). *Quantum GIS User Guide*. Online verfügbar unter [http://download.osgeo.org/qgis/doc/manual/qgis-1.6.0\\_user\\_guide\\_en.pdf](http://download.osgeo.org/qgis/doc/manual/qgis-1.6.0_user_guide_en.pdf) (Abrufdatum 10.04.2012). Open Source Geospatial Foundation.
- Radke, R. J. u. a. (2005). „Image change detection algorithms: a systematic survey“. In: *IEEE Transactions on Image Processing, ITIP* 14.3, S. 294–307.
- Ramer, U. (1972). „An iterative procedure for the polygonal approximation of plane curves“. In: *Computer Graphics and Image Processing* 1.3, S. 244–256.
- Rasband, W. S. (1997). „ImageJ - Image Processing and Analysis in Java“. In: 2012. Online verfügbar unter <http://rsb.info.nih.gov/ij/> (Abrufdatum 27.04.2012).
- Raza, A. und W. Kainz (2002). „An object-oriented approach for modeling urban land-use changes“. In: *URISA Journal* 14.1, S. 37–56.
- Rees, G. und W. G. Rees (2012). *Physical principles of remote sensing*. Cambridge University Press.
- Richards, J. A. (2013). *Remote sensing digital image analysis: an introduction*. Springer.
- Rodd, E. (1972). *Closed Boundary Field Selection in Multi-spectral Digital Images*. IBM.
- Rosenberg, C. (1996). *The Lenna Story*. Online verfügbar unter [www.lenna.org](http://www.lenna.org) (Abrufdatum 03.07.2012).
- Ruefenacht, B. u. a. (2002). *New Techniques for Segmenting Images*. Techn. Ber. Remote Sensing Applications Center, Salt Lake City.

- Ryherd, S. und C. Woodcock (1996). „Combining spectral and texture data in the segmentation of remotely sensed images“. In: *Photogrammetric Engineering Remote Sensing* 62.2, S. 181–194.
- Saalfeld, A. (1999). „Topologically consistent line simplification with the Douglas-Peucker algorithm“. In: *Cartography and Geographic Information Science* 26.1, S. 7–18.
- Sahoo, P. K. u. a. (1988). „A survey of thresholding techniques“. In: *Computer vision, graphics, and image processing* 41.2, S. 233–260.
- Schiewe, J. (2002). „Segmentation of high-resolution remotely sensed data-concepts, applications and problems“. In: *Proceedings of the ISPRS Commission IV Symposium "Geospatial Theory, Processing and Applications"*. Bd. XXXIV Part 4. Natural Resources Canada, S. 380–385.
- Schiewe, J. u. a. (2001). „Potential and problems of multi-scale segmentation methods in remote sensing“. In: *GeoBITGIS* 49, S. 34–39.
- Schöpfer, E. und S. Lang (2006). „Object fate analysis-a virtual overlay method for the categorisation of object transition and object-based accuracy assessment“. In: *Proceedings of the OBIA 2006 Conference - 1st International Conference on Object-based Image Analysis*.
- Schwartzkopf, W. u. a. (2002). „Entropy estimation for segmentation of multi-spectral chromosome images“. In: *Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, S. 234–237.
- Sezgin, M. und B. Sankur (2001). „Selection of thresholding methods for nondestructive testing applications“. In: *Proceedings of the International Conference on Image Processing, ICIP*. Bd. 3.
- Shi, J. und J. Malik (2000). „Normalized cuts and image segmentation“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, ITPAMI* 22.8, S. 888–905.
- Singaraju, D. u. a. (2009). „P-brush: Continuous valued MRFs with normed pairwise distributions for image segmentation“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, ICVPR*. IEEE, S. 1303–1310.
- Singh, A. (1989). „Review Article Digital change detection techniques using remotely-sensed data“. In: *International Journal of Remote Sensing* 10.6, S. 989–1003.
- Skarbek, W. und A. Koschan (1994). „Colour image segmentation: A survey“. In:
- Skurikhin, A. (2008). „Proximity Graphs Based Multi-scale Image Segmentation“. In: *Advances in Visual Computing*. Hrsg. von G. Bebis u. a. Bd. 5358. Lecture Notes in Computer Science. Springer, S. 298–307.
- Solutions, ITT Visual Information (2008). *ENVI Feature Extraction Module User's Guide*.
- Sonka, M. u. a. (2008). *Image Processing, Analysis, and Machine Vision, International Student Edition*. Third. Thomson.
- Stow, D. A. (2010). „Handbook of Applied Spatial Analysis“. In: Springer. Kap. Geographic object-based image change analysis, S. 565–582.
- Stow, D. A. u. a. (2008). „Monitoring shrubland habitat changes through object-based change identification with airborne multispectral imagery“. In: *Remote Sensing of Environment* 112.3, S. 1051–1061.
- Strahler, A. H. u. a. (1980). „Incorporating collateral data in Landsat classification and modeling procedures“. In: *Proceedings of the 14th International Symposium on Remote Sensing of Environment, ISRSE*, S. 1009–1026.
- The White House Office of the Press Secretary (Sep. 2009). *Statements by President Obama, French President Sarkozy, and British Prime Minister Brown on Iranian Nuclear Facility*. Online verfügbar unter [www.whitehouse.gov](http://www.whitehouse.gov) (Abrufdatum 22.05.2013).
- Tilton, J. C. (2003). „Analysis of hierarchically related image segmentations“. In: *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, S. 60–69.
- Treaty on the Non-Proliferation of Nuclear Weapons* (Juli 1968). Online verfügbar unter <http://www.armscontrol.org/documents/npt> (Abrufdatum 01.09.2013).

- Tu, Z. und S. C. Zhu (2002). „Image segmentation by data-driven Markov chain Monte Carlo“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, ITPAMI* 24.5, S. 657–673.
- Tu, Z. u. a. (2005). „Image parsing: Unifying segmentation, detection, and recognition“. In: *International Journal of Computer Vision* 63.2, S. 113–140.
- Ueda, N. u. a. (2000). „SMEM algorithm for mixture models“. In: *Neural computation* 12.9, S. 2109–2128.
- Unnikrishnan, R. u. a. (2007). „Toward objective evaluation of image segmentation algorithms“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, ITPAMI* 29.6, S. 929–944.
- Van Coillie, F. M. B. u. a. (2008a). „Semi-automatic forest stand delineation using wavelet based segmentation of very high resolution optical imagery“. In: *Object-based image analysis: spatial concepts for knowledge-driven remote sensing applications*, S. 237.
- Van Coillie, F. u. a. (2008b). „Quantitative segmentation evaluation for large scale mapping purposes“. In: *Proceedings of the GEOBIA 2008 Conference: Pixels Objects Intelligence Geographic Object Based Image Analysis for the 21st Century*.
- Walter, V. (2004). „Object-based classification of remote sensing data for change detection“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 58.3-4, S. 225–238.
- Wertheimer, M. (1923). „Untersuchungen zur Lehre von der Gestalt. II“. In: *Psychological Research* 4.1, S. 301–350.
- Wiemker, R. (1997). „An iterative spectral-spatial Bayesian labeling approach for unsupervised robust change detection on remotely sensed multispectral imagery“. In: *Computer Analysis of Images and Patterns*. Springer, S. 263–270.
- Wikipedia (2012). *Gradientenverfahren* — *Wikipedia, Die freie Enzyklopädie*. Online verfügbar unter <http://de.wikipedia.org/w/index.php?title=Gradientenverfahren&oldid=99783208> (Abrufdatum 5. August 2012).
- Willhauck, G. u. a. (2000). „Comparison of object oriented classification techniques and standard image analysis for the use of change detection between SPOT multispectral satellite images and aerial photos“. In: *Proceedings of the XIXth ISPRS Congress*. Bd. XXXIII, part B3 Supplement. Citeseer, S. 35–42.
- Williams, D. (März 2011). *U.S. laws keep "shutter control" over images of Israel*. Online verfügbar unter <http://www.reuters.com/article/2011/03/10/israel-turkey-satellites-idUSLDE7260HS20110310> (Abrufdatum 28.06.2011).
- Wu, S.-T. und M. R. G. Marquez (2003). „A non-self-intersection Douglas-Peucker algorithm“. In: *Proceedings of the XVIth Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI*, S. 60–66.
- Wu, Z. und R. Leahy (1993). „An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, ITPAMI* 15.11, S. 1101–1113.
- Xiaochun, L. und C. Jing (2004). „An algorithm for SAR image segmentation“. In: *Proceedings of the 4th International Conference on Microwave and Millimeter Wave Technology, ICMWT*, S. 647–650.
- Xie, E. u. a. (2011). „GPU implementation of spiking neural networks for color image segmentation“. In: *4th International Congress on Image and Signal Processing, CISP*. Bd. 3, S. 1246–1250.
- Yang, A. Y. u. a. (2008). „Unsupervised segmentation of natural images via lossy data compression“. In: *Computer Vision and Image Understanding* 110.2, S. 212–225.
- Zhang, H. u. a. (2004). „An entropy-based objective evaluation method for image segmentation“. In: *Proceedings of the SPIE Storage and Retrieval Methods and Applications for Multimedia Conference* 2.4.

- Zhang, H. u. a. (2008). „Image segmentation evaluation: A survey of unsupervised methods“. In: *Computer Vision and Image Understanding* 110.2, S. 260–280.
- Zhou, W. u. a. (2008). „Object-based land cover classification and change analysis in the Baltimore metropolitan area using multitemporal high resolution remote sensing data“. In: *Sensors* 8.3, S. 1613–1636.
- Zoller, T. und J. M. Buhmann (2007). „Robust image segmentation using resampling and shape constraints“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, ITPAMI* 29.7, S. 1147–1164.

Band / Volume 228

**Implementation of capillary electromigrative separation techniques coupled to mass spectrometry in forensic and biological science**

T. N. Posch (2014), 264 pp

ISBN: 978-3-89336-987-4

Band / Volume 229

**Mobile Messungen - Messung und Bewertung von Verkehrsemissionen**

C. Ehlers (2014), VII, 136 pp

ISBN: 978-3-89336-989-8

Band / Volume 230

**Data Processing and Trace Gas Retrievals for the GLORIA Limb Sounder**

T. Guggenmoser (2014), xi, 111, XXVI

ISBN: 978-3-89336-993-5

Band / Volume 231

**Assessment of Laser Induced Ablation Spectroscopy (LIAS) as a method for quantitative in situ surface diagnostic in plasma environments**

N. Gierse (2014), 199 pp

ISBN: 978-3-89336-994-2

Band / Volume 232

**Lattice-Boltzmann-Simulation in faserbasierten Mikrostrukturen**

J. P. Brinkmann (2014), viii, 129 pp

ISBN: 978-3-89336-995-9

Band / Volume 233

**Numerische Simulationen von Abfallgebinden aus der Wiederaufarbeitung von Kernbrennstoffen**

S. Schneider (2014), 201 pp

ISBN: 978-3-89336-998-0

Band / Volume 234

**Materials for Advanced Power Engineering 2014**

J. Lecomte-Beckers, O. Dedry, J. Oakey, B. Kuhn (Ed.)  
(2014), xxi, 930, viii pp

ISBN: 978-3-95806-000-5

Band / Volume 235

**Untersuchungen zur Abtrennung, Konversion und Transmutation von langlebigen Radionukliden**

Ein Beitrag zur fortschrittlichen Entsorgung von hochradioaktiven Abfällen

G. Modolo (2014), 291 pp

ISBN: 978-3-95806-005-0



Band / Volume 236

**Keramische Membranen für die H<sub>2</sub>-Abtrennung in CO-Shift-Reaktoren**

D. van Holt (2014), IV, 169 pp

ISBN: 978-3-95806-007-4

Band / Volume 237

**Entwicklung von porösen Silica-Membranen zur CO<sub>2</sub>-Abtrennung aus dem Rauchgas fossil befeuerter Kraftwerke**

J. S. Eiberger (2014), II, 163 pp

ISBN: 978-3-95806-008-1

Band / Volume 238

**Development of a highly sensitive and versatile mass spectrometer system for laboratory and atmospheric measurements**

S. Albrecht (2014), iv, 171 pp

ISBN: 978-3-95806-009-8

Band / Volume 239

**High Temperature Corrosion of Alloys and Coatings in Gas-Turbines Fired with Hydrogen-Rich Syngas Fuels**

W. Nowak (2014), 201 pp

ISBN: 978-3-95806-012-8

Band / Volume 240

**Einfluss neuer Membranmaterialien auf Herstellung und Betrieb von Direktmethanol-Brennstoffzellen**

S. Hürter (2014), V, 164 pp

ISBN: 978-3-95806-013-5

Band / Volume 241

**Spannungsinduziertes Versagen in Hochtemperaturschichtsystemen**

C. Nordhorn (2014), v, 118 pp

ISBN: 978-3-95806-016-6

Band / Volume 242

**Änderungsdetektion digitaler Fernerkundungsdaten mittels objekt-basierter Bildanalyse**

C. Listner (2014), 176 pp

ISBN: 978-3-95806-017-3

Weitere **Schriften des Verlags im Forschungszentrum Jülich** unter  
<http://wwwzb1.fz-juelich.de/verlagextern1/index.asp>





**Energie & Umwelt / Energy & Environment**  
**Band / Volume 242**  
**ISBN 978-3-95806-017-3**

